

UFRRJ
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM
MODELAGEM MATEMÁTICA E COMPUTACIONAL

DISSERTAÇÃO

**Modelos de otimização combinatória aplicados à
gestão de demandas operacionais em centros
universitários**

Amanda Moreira de Medeiros Brasil

2023



**UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM
MATEMÁTICA E COMPUTACIONAL**

**MODELOS DE OTIMIZAÇÃO COMBINATÓRIA APLICADOS À
GESTÃO DE DEMANDAS OPERACIONAIS EM CENTROS
UNIVERSITÁRIOS**

AMANDA MOREIRA DE MEDEIROS BRASIL

Sob orientação de
Ronaldo Malheiros Gregório

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre** no Programa de Pós-Graduação em Modelagem Matemática e Computacional, Área de Concentração em Modelagem Matemática e Computacional.

Seropédica, RJ, Brasil
Dezembro de 2023

Universidade Federal Rural do Rio de Janeiro
Biblioteca Central / Seção de Processamento Técnico

Ficha catalográfica elaborada
com os dados fornecidos pelo(a) autor(a)

B823m Brasil, Amanda Moreira de Medeiros, 1991-
Modelos de otimização combinatória aplicados à gestão
de demandas operacionais em centros universitários /
Amanda Moreira de Medeiros Brasil. - Duque de Caxias,
2023.
57 f.

Orientador: Ronaldo Malheiros Gregório.
Dissertação(Mestrado). -- Universidade Federal Rural
do Rio de Janeiro, Programa de Pós-Graduação em
Modelagem Matemática e Computacional, 2023.

1. Otimização. 2. Modelagem Matemática
computacional. I. Gregório, Ronaldo Malheiros, 1978-
orient. II Universidade Federal Rural do Rio de
Janeiro. Programa de Pós-Graduação em Modelagem
Matemática e Computacional III. Título.



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS



PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

Seropédica-RJ, 21 de dezembro de 2023.

Amanda Moreira de Medeiros Brasil

Dissertação submetida como requisito parcial para a obtenção de grau de **Mestra**, no Programa de Pós-Graduação em Modelagem Matemática e Computacional PPGMMC, área de Concentração em Modelagem Matemática e Computacional.

DISSERTAÇÃO APROVADA EM 21/12/2023

Ronaldo Malheiros Gregório Dr^o UFRRJ (Orientador, Presidente da Banca)

Marcelo Dib Cruz Dr^o UFRRJ (membro interno)

Renan Vicente Pinto Dr^o UFRRJ (Externo ao Programa)

Luidi Gelaberti Simonetti Dr^o UFRJ (Externo à Instituição)



ATA N° ata/2023 - ICE (12.28.01.23)

(N° do Documento: 6725)

(N° do Protocolo: NÃO PROTOCOLADO)

(Assinado digitalmente em 30/12/2023 11:08)

MARCELO DIB CRUZ

PROFESSOR DO MAGISTERIO SUPERIOR

DCOMP (11.39.97)

Matrícula: ###680#1

(Assinado digitalmente em 28/12/2023 13:39)

RENAN VICENTE PINTO

PROFESSOR DO MAGISTERIO SUPERIOR

DeptM (12.28.01.00.00.63)

Matrícula: ###812#3

(Assinado digitalmente em 27/12/2023 13:03)

RONALDO MALHEIROS GREGORIO

PROFESSOR DO MAGISTERIO SUPERIOR

DeptTL/IM (12.28.01.00.00.90)

Matrícula: ###696#7

(Assinado digitalmente em 27/12/2023 11:22)

LUIDI GELABERT SIMONETTI

ASSINANTE EXTERNO

CPF: ###.###.927-##

Visualize o documento original em <https://sipac.ufrrj.br/documentos/> informando seu número: 6725, ano: 2023, tipo: ATA, data de emissão: 27/12/2023 e o código de verificação: cf97f46758

UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

AMANDA MOREIRA DE MEDEIROS BRASIL

Dissertação submetida como requisito parcial para obtenção do grau de **Mestre** no Programa de Pós-Graduação em Modelagem Matemática e Computacional, Área de Concentração em Modelagem Matemática e Computacional.

21/12/2023.

Ronaldo Malheiros Gregório, D. Sc. UFRRJ
(Presidente)

Marcelo Dib Cruz, D. Sc. UFRRJ

Renan Vicente Pinto, D. Sc. UFRRJ

Luidi Gelabert Simonetti, D. Sc. UFRJ

Agradecimentos

“O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.”

*"Quem tem um porquê, enfrenta
qualquer como"
(Viktor Frankl)*

DE MEDEIROS BRASIL, Amanda Moreira. **Modelos de otimização combinatória aplicados à gestão de demandas operacionais em centros universitários**. 2023. 57f. Dissertação (Mestrado em Modelagem Matemática e Computacional). Instituto de Ciências Exatas, Universidade Federal Rural do Rio de Janeiro, Seropédica, RJ, 2023.

A gestão operacional em instituições públicas de ensino e pesquisa, como os centros universitários, representa um grande desafio para os gestores educacionais, muitas vezes desviando o foco das questões pedagógicas relacionadas à produção de conhecimento científico e tecnológico. O objetivo desta proposta é apresentar modelos de apoio ao gerenciamento da distribuição de disciplinas de diferentes cursos de graduação por salas de aula e ao gerenciamento de cardápios providos por restaurantes em centros universitários, bem como conduzir simulações computacionais com cenários de médio/grande porte que aproximam o cotidiano dessas instituições. Para abordar esses problemas, são considerados modelos clássicos de otimização linear inteira binária e linear inteira mista. As soluções para esses modelos são sintetizadas usando métodos de enumeração implícita e planos de corte disponíveis na biblioteca *cPlex* para Python.

Palavras-chave: Modelagem matemática, otimização linear inteira mista, problema de alocação de salas de aula, problema da dieta.

ABSTRACT

DE MEDEIROS BRASIL, Amanda Moreira. **Combinatorial optimization models applied to the management of operational demands in university centers.** 2023. 57p. Dissertation (Master in Mathematical and Computational Modeling). Instituto de Ciências Exatas, Universidade Federal Rural do Rio de Janeiro, Seropédica, RJ, 2023.

Operational management in public teaching and research institutions, such as university centers, represents a major challenge for educational managers, often diverting the focus from pedagogical issues related to the production of scientific and technological knowledge. The objective of this proposal is to present models to support the management of the distribution of subjects from different undergraduate courses across classrooms and the management of menus provided by restaurants in university centers, as well as to conduct computer simulations with medium/large scenarios that approximate the daily life of these institutions. To address these problems, classical binary integer linear and mixed integer linear optimization models are considered. Solutions to these models are synthesized using implicit enumeration methods and cutting planes available in the PuLP library for Python.

Keywords: Mathematical modeling, mixed integer linear optimization, classroom assignment problem, diet problem.

1	INTRODUÇÃO	1
1.1	Aspectos gerais e abrangência da pesquisa	1
1.2	O modelo da dieta	2
1.3	O problema de alocação de sala de aula	3
1.4	Estrutura do trabalho	4
2	O MODELO PRIMAL DE PROGRAMAÇÃO LINEAR	6
2.1	O modelo primal	6
2.2	Soluções básicas viáveis	8
2.3	O Teorema Fundamental da Programação Linear	10
2.4	A geometria de um <i>PPL</i>	13
2.5	Algoritmo Simplex primal	14
3	O MODELO DUAL DE PROGRAMAÇÃO LINEAR	19
3.1	Definição e exemplos	19
3.2	Aspectos gerais da teoria da dualidade	21
3.3	Algoritmo Simplex dual	24
3.4	Algoritmo simplex primal-dual	26
4	PROGRAMAÇÃO LINEAR INTEIRA	30
4.1	Programação Linear Inteira	30
4.1.1	Definição e aspectos gerais de otimização linear inteira	30
4.2	Modelos lineares inteiros binários e mistos	31
4.2.1	Modelos binários	32
4.2.2	Modelos inteiros mistos	33
4.3	Método de enumeração e planos de corte	34
4.3.1	Métodos de enumeração - aspectos gerais	34
4.3.2	Planos de corte - aspectos gerais	36
4.3.3	Branch and Bound	37
4.3.4	Método de Gomory	37
4.3.5	Branch and cut	38
4.3.6	Benders	39

5	MODELOS DE GESTÃO OPERACIONAL DE RESTAURANTES E DE SALAS DE AULA EM CENTROS UNIVERSITÁRIOS E SIMULAÇÕES COMPUTACIONAIS	41
5.1	Modelo de gestão operacional de restaurantes universitários	41
5.2	Modelo de gestão de salas de aula em centros universitários	44
5.3	Descrição de cenário para o modelo de gestão de restaurantes universitários aplicado ao RU-IM/UFRRJ	47
5.4	Descrição de cenário para o modelo de gestão de salas de aula em centros universitários aplicado ao IM/UFRRJ	49
5.5	Simulações computacionais	52
5.5.1	Simulação computacional para a gestão operacional do RU-IM	52
5.5.2	Simulação computacional para a gestão de salas de aula do Instituto Multidisciplinar	53
6	CONCLUSÕES E TRABALHOS FUTUROS	55
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	56

Lista de Figuras

Figura 3.1 – Gráfico do valor ótimo.....	25
Figura 4.1 – Restrições de cardinalidade para o problema do caixeiro viajante. Extraído de [2], em 09/12/2023.....	35
Figura 4.2 – Problema da mochila no Python	36
Figura 4.3 – Taxonomia para problemas de fluxo.Extraído de [2], em 09/12/2023	40

Lista de Abreviações e Siglas

UFRRJ	: UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO
RJ	: RIO DE JANEIRO
PL	: PROBLEMA LINEAR
PPL	: PROBLEMA DE PROGRAMAÇÃO LINEAR
LI	: LINEARMENTE INDEPENDENTES
LD	: LINEARMENTE DEPENDENTES
RU	: RESTAURANTE UNIVERSITÁRIO
PAS	: PROBLEMA DE ALOCAÇÃO DE SALA

1.1 Aspectos gerais e abrangência da pesquisa

Os problemas de otimização linear inteira binária e linear inteira mista ganharam notoriedade na década de 1940, quando vários pesquisadores propuseram modelos para diversas aplicações. Um dos exemplos mais significativos foi o problema da dieta durante a Segunda Guerra Mundial, Conforme destacado em (Dantzing, 2002) [3]. Segundo (Wolsey, 1998) [4], (Arenales *et al*, 2015) [5] e (Goldberg and Luna, 2005)[2], existem modelos clássicos para ambas as formulações, como os problemas da *mochila*, *corte*, *atribuição*, *cobertura*, *agrupamento*, *empacotamento* e *caixeiro-viajante*. Além disso, tais formulações descrevem aplicações modernas, como os problemas de *programação de horários de trens*, *alocação de tripulações aéreas*, *planejamento de produção*, em particular, *planejamento de produção de eletricidade*, *telecomunicações*, *rotas e escalas de veículos para transporte de pessoas* e *programação de chegadas e partidas de viagens aéreas*.

Com o desenvolvimento do algoritmo Simplex, fundamentado nos avanços iniciais do matemático russo Leonard Kantorovich e estabelecido por Dantzig, e com os progressos tecnológicos impulsionados pela introdução de computadores para realizar os cálculos inerentes ao algoritmo, os modelos de programação linear passaram a ser resolvidos de forma eficiente e em tempo factível, como apontado por [5]. Isso representou um avanço significativo não apenas para a resolução do modelo da dieta, que simplificada figura como um modelo de programação linear, mas também permitiu o refinamento de técnicas duais sofisticadas para estabelecer limites para os valores ótimos de modelos de programação linear inteira. De fato, os métodos duais baseados em relaxações lineares e lagrangianas do modelo original de programação inteira tornaram-se as principais técnicas utilizadas para determinar soluções aproximadas para problemas dessa natureza, como mencionado por [4] e [5].

Neste trabalho, são apresentadas formulações de otimização linear inteira 0-1 e linear inteira mista para resolver duas demandas operacionais em centros universitários. Baseando-se em duas formulações genéricas, os modelos abordam os desafios de fornecimento de refeição aos discentes e alocação de disciplinas em salas de aula, estendendo o modelo apresentado em (Ferreira *et al*, 2021)[6], através da introdução de variáveis que representam porções inteiras para determinados itens alimentícios e refinando o modelo apresentado em (Rangel *et al*, 2023) [7]. As modelagens propostas simplificam linearmente ambas as demandas, a alocação semanal de disciplinas em salas de aula e o fornecimento semanal de refeições aos alunos.

Embora o problema da dieta seja comumente apresentado como um modelo linear puro em sua formulação mais propagada, a realidade é que ele engloba características inteiras. Isso se deve ao fato de que diversos alimentos só podem ser fornecidos em porções completas. Al-

guns exemplos são frutas tropicais, como banana, maçã, laranja, entre outras, bem como cortes de carne suína ou bovina, peixe e frango, que são oferecidos em porções inteiras como fonte de nutrientes. Por outro lado, para alocação de salas de aula, são consideradas restrições que incorporam percentagens de disciplinas por departamentos que podem ser atribuídas a turnos diferentes dos originalmente requeridos. Essas perspectivas destacam a contribuição do presente trabalho em relação às modelagens apresentadas em [7] e [6]. Além disso, realizamos simulações computacionais para ambos os modelos de otimização sintetizados. Neste estudo, as simulações foram realizadas em cenários similares aos encontrados no *campus* Nova Iguaçu da Universidade Federal Rural do Rio de Janeiro para facilitar a análise dos resultados obtidos.

1.2 O modelo da dieta

O modelo de dieta, também conhecido como problema da mistura, é um exemplo clássico de Problema de otimização linear que foi inicialmente concebido por George Joseph Stigler na década de 1930. Ele surgiu com o objetivo primordial de criar uma combinação de alimentos, especialmente rações, para os soldados em zonas de combate, visando minimizar os custos de produção e o volume de carga que precisavam carregar durante um combate. A principal aplicação do modelo de Stigler ocorreu durante a Segunda Guerra Mundial (1941-1945). Além da redução dos custos econômicos das rações dos soldados e da minimização do volume que transportavam, havia também uma preocupação essencial com a nutrição básica, conforme visto em (Buttriss *et al*, 2014) [8], [3] e (Dooren, 2018) [9].

Conforme orientações dos órgãos de saúde, era crucial garantir uma alimentação que fornecesse energia, proteínas, minerais e vitaminas em quantidade suficiente para manter os soldados bem nutridos e ativos durante todas as batalhas. Isso visava capacitar os soldados a lidar com os desafios militares da melhor maneira possível, além de estabelecer estratégias e táticas de guerra eficazes. Inicialmente, a solução para o problema da dieta não foi prontamente alcançada, levando muitos matemáticos e acadêmicos a se dedicarem a essa questão, dada a perspectiva inovadora que tal solução representaria para os conflitos armados a partir deste ponto.

Em 1947, após a guerra, George Danzig propôs o método Simplex [3], um algoritmo que resolve não apenas o problema de Stigler, mas também problemas de Otimização Linear de uma maneira geral. Depois disso, vários aprimoramentos foram incorporados ao algoritmo considerando a teoria de dualidade, gerando variantes eficientes do método, como os algoritmos Simplex dual e primal-dual (Bazaraa *et al*, 2010) [10], (Luenberger and Ye, 2016) [11], (Vanderbei, 2013)[12].

Diversos pesquisadores utilizaram o modelo da dieta para estruturar o planejamento de refeições oferecidas a estudantes, desde o ensino fundamental e médio até instituições universitárias. Esse método visa prever as porções de alimentos a serem disponibilizadas em cada refeição, proporcionando aos alunos nutrientes ao longo dos dias e minimizando os custos, conforme destacado por (Baki *et al*) [13]. No entanto, a obtenção manual dessas soluções por meio do método Simplex torna-se inviável à medida que o número de variáveis envolvidas no problema aumenta, resultando em um crescimento exponencial do número de equações. Assim, a síntese de soluções para problemas de otimização linear tornou-se viável apenas com avanços tecnológicos, especialmente o desenvolvimento de computadores mais rápidos e softwares mais sofisticados. Na verdade, em 1947, a equipe de Dantzig levou várias semanas para resolver o Problema da Dieta, apesar de contar com vários colaboradores equipados com calculadoras que os auxiliaram nos cálculos. Existem algumas aplicações do modelo da dieta no cotidiano de empresas, uma delas exposta em [12], se apresenta de acordo com o seguinte contexto.

Considere uma empresa farmacêutica que produza em forma de comprimido cada um dos nutrientes considerados importantes pelo nutricionista. A empresa farmacêutica tenta convencer o nutricionista a comprar comprimidos e, assim, fornecer os nutrientes diretamente, em vez da compra de vários alimentos. O problema da farmacêutica consiste em determinar preços unitários positivos $\lambda_1, \lambda_2, \dots, \lambda_m$ para os m nutrientes, de modo a maximizar a receita e, ao mesmo tempo, ser competitivo com alimentos reais. Mas, para ser competitivo com alimentos, o custo de uma unidade do comprimido produzido sinteticamente a partir de nutrientes puros, fornecidos pela farmacêutica, não deve ser maior que o preço de mercado dos alimentos que fornecem o mesmo nutriente.

Neste estudo, o modelo de dieta foi adaptado para otimizar a gestão de restaurantes universitários. O objetivo é minimizar os custos financeiros associados à compra de insumos, reduzindo ao mesmo tempo o desperdício, ao mesmo tempo em que se garante uma oferta diária de refeições balanceadas. Essas refeições devem atender às necessidades nutricionais semanais dos estudantes.

Ao buscar a minimização do desperdício, é essencial considerar os diferentes períodos de oferta das refeições (manhã, tarde ou noite). Isso permite estimar o número de estudantes a serem atendidos em cada turno, baseando-se na alocação das disciplinas por salas de aula para esses períodos e, conseqüentemente, na quantidade de estudantes presentes.

O problema da dieta, com aplicação à cenários reais, já foi alvo de diversos estudos em trabalhos presentes na literatura científica. As abordagens apresentadas em [13] e [9] exemplificam essa afirmação. No estudo conduzido por [13], a pesquisa foi realizada com base nas observações de 100 alunos de um centro universitário que ajudaram a compreender os padrões alimentares de cada estudante, especificamente no café da manhã, almoço e jantar. Para abordar o problema da dieta, a pesquisa focou em seis nutrientes. São eles: proteína, gordura, carboidratos, cálcio, fósforo e vitamina C. Contudo, a concepção de uma dieta que satisfaça a ingestão ideal de todos os nutrientes em uma única refeição não se adapta facilmente à realidade. Na verdade, desenvolver um único modelo de programação linear que atenda às necessidades nutricionais individuais de cada aluno, uma vez que as preferências e demandas nutricionais variam consideravelmente de pessoa para pessoa não é uma representação fidedigna da realidade.

Abordagens práticas baseadas no modelo da dieta geralmente consideram grupos de pessoas com características semelhantes, como faixa etária e massa corporal dentre outras. Essas abordagens incluem limites diários de nutrientes e calorias, adaptando-se melhor às necessidades coletivas de um grupo específico.

1.3 O problema de alocação de sala de aula

O problema de alocação de salas de aula é um desafio frequentemente enfrentado pelas instituições de ensino (Schaerf, 1999) [14]. Este problema pode agravar-se ainda mais quando são estabelecidas algumas restrições, como a capacidade das salas, a utilização de salas com equipamentos específicos, por exemplo, laboratórios, dentre outros aspectos. Na maioria dos casos, este problema é resolvido manualmente pelas equipas de administração da instituição. Por ser um problema NP-difícil, uma solução viável pode exigir muito tempo e esforço por parte dos responsáveis (Elloumi *et al*) [15] para ser computada. Além disso, corre-se ainda o risco de gerar insatisfação de alunos e professores e perturbações no cotidiano das instituições caso uma solução viável, porém não ótima, seja operacionalizada.

Na literatura encontram-se vários modelos relacionados com a alocação de disciplinas em sala de aula de cursos universitários. Por exemplo, (Bucco *al*, 2017) [16] aplicou tal formulação à vários cenários. Além disso, destacam-se ainda os modelos apresentados para a Virginia

Tech (EUA) (Sarin *et al*, 2010) [17], o sistema universitário espanhol, especificamente, a Universidade de Valência (Alvarez-Valdes *et al*, 2002) [18], a Universidade de Purdue (Rudová *et al*, 2011) [19], a Università degli studi di Udine (Burke *et al*, 2009) [20], o sistema de educação Grego (Daskalaki *et al*, 2004) [21] e para a Universidade de Sydney (Beyrouthy *et al*, 2009) [22].

O problema da atribuição de salas de aula possui estrutura similar ao modelo de atribuição de tarefas, pois envolve a delegação de tarefas à agentes. Se considerássemos que os sujeitos assumem o papel de tarefas e as salas de aula, o papel de agentes, o problema de alocação de salas de aula se configuraria como um caso particular do problema de atribuição de tarefas. Porém a quantidade de disciplinas deveria ser igual à quantidade de salas de aula, o que geralmente não é verdade. Na prática, o número de disciplinas é muito superior ao número de horários disponíveis para atribuição, o que aumenta a sua complexidade. É necessário utilizar um critério de decisão para escolher a disciplina que será atribuída em primeiro lugar, caso duas disciplinas estejam listadas para serem atribuídas à mesma sala de aula.

O modelo de alocação de salas de aula também é utilizado na construção de horários para a marcação de serviços de veículos, alocação de tripulações em voos, entre outras aplicações. Uma característica do problema de alocação de salas de aula é que ele também depende das particularidades da instituição analisada, uma vez que os centros universitários geralmente dispõem de instalações diferentes de acordo com a disponibilidade de seus recursos, necessidades e exigências.

O problema de alocação de salas consiste na alocação de disciplinas, com horários já estabelecidos, a salas de aulas, considerando a capacidade da sala e as necessidades dos docentes e discentes. Esse processo, geralmente, é resolvido pelas instituições de ensino manualmente o que além de levar vários dias para ser concluído, muitas vezes não garante a alocação eficiente dos espaços. Essa situação também é enfrentada pelo Instituto Multidisciplinar da UFRRJ que conta com cerca de 11 cursos de graduação funcionando nos três turnos (manhã/tarde/noite), além dos cursos de pós-graduação, com cada disciplina tendo sua necessidade, seja pela estrutura da sala, solicitação do docente quanto ao horário da disciplina, ou então a quantidade de discentes inscritos na disciplina.

1.4 Estrutura do trabalho

A estrutura deste trabalho destaca, no Capítulo 2, tópicos relacionados ao modelo primal de otimização linear, bem como detalhes sobre o algoritmo Simplex em sua forma primal. Os conceitos e resultados destacados guardam relação estrita com os objetivos da pesquisa e viabilizam a compreensão de algoritmos de enumeração implícita e planos de corte em otimização inteira. A descrição completa do algoritmo Simplex primal pode ser encontrada em [11] e [12]. No Capítulo 3, são destacados conceitos teóricos sobre o modelo dual de Programação Linear, além de detalhes sobre o algoritmo Simplex em suas formas dual e primal-dual. No Capítulo 4, a Programação Linear Inteira 0-1 e a Mista é abordada de maneira introdutória, onde são apresentadas definições e exemplos, bem como uma breve introdução aos principais métodos de enumeração implícita e planos de corte, tendo em vista que os pacotes computacionais utilizados para sintetizar soluções para os modelos apresentados se baseiam em tais metodologias. No capítulo 5, apresentamos os modelos de gestão operacional de restaurantes universitários, que destaca a formulação de cardápios a serem servidos aos discentes, e de gestão de espaço físico, mais especificamente, de salas de aula. Além disso, descrevemos os cenários para ambos os modelos no *campus* Nova Iguaçu da Universidade Federal Rural do Rio de Janeiro. O Capítulo

5 também oferece uma análise das soluções obtidas, propiciando uma visão detalhada sobre as estratégias de otimização empregadas, os critérios de desempenho considerados e as conclusões derivadas das simulações. Ao examinar os desdobramentos e implicações práticas desses modelos, pretendemos oferecer uma compreensão abrangente de suas aplicações e relevância em contextos de alocação de recursos e planejamento eficiente.

O modelo primal de programação linear

2.1 O modelo primal

Esta seção aborda elementos da teoria de otimização linear. As notações, definições e resultados foram adaptados de [10], [11] e [12]. Aspectos relacionados a degenerescência de soluções básicas não são tratados neste trabalho. Dessa forma, recomendamos [12] para estudos mais aprofundados acerca do tema.

Nos problemas abordados na literatura, existem variáveis cujos valores devem ser escolhidos de forma a otimizar (maximizar ou minimizar) o o objetivo do problema. Estas variáveis são denominadas *variáveis de decisão* e geralmente são denotadas por:

$$x_j, \quad j = 1, 2, \dots, n$$

Na programação linear, o objetivo é sempre maximizar ou minimizar alguma função linear que depende dessas variáveis de decisão, à saber, um objetivo representado matematicamente por:

$$z = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

Esta função é denominada função objetivo. Muitas vezes parece que os problemas do mundo real são mais naturalmente formulados como problemas cujo objetivo é a minimização. Porém, na prática, alguns problemas também se apresenta sob a forma de maximização. Quando há a necessidade de conversão de problemas de maximização para minimização basta multiplicar a função objetivo por -1. Note que maximizar z equivale a minimizar $-z$, multiplicando o valor ótimo do problema de minimização por -1 para obter o valor ótimo correspondente ao problema original, de maximização.

Além da função objetivo, os problemas também possuem restrições, algumas simples e outras mais complexas. Contudo a exigência canônica é que, de uma maneira geral, as variáveis sejam não negativas. Além disso, as restrições não canônicas podem figurar sob a forma de igualdades ou desigualdades lineares.

Suponha que um problema de programação linear contenha um total de m restrições não canônicas. Em geral, elas podem se apresentar sob uma das formas sintetizadas a seguir:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \left\{ \begin{array}{l} \leq \\ = \\ \geq \end{array} \right\} b_i, 1 \leq i \leq m.$$

O processo para converter uma restrição de desigualdade, suponhamos a l -ésima, para algum $1 \leq l \leq m$, da forma \leq , em igualdade é simples.

$$a_{l1}x_1 + a_{l2}x_2 + \dots + a_{ln}x_n \leq b_l,$$

Basta então adicionar uma variável não negativa, w , denominada variável de folga, como é mostrado a seguir:

$$a_{l1}x_1 + a_{l2}x_2 + \cdots + a_{ln}x_n + w = b_l, \quad w \geq 0$$

Por outro lado, se a l -ésima for da forma \geq ,

$$a_{l1}x_1 + a_{l2}x_2 + \cdots + a_{ln}x_n \geq b_l,$$

e desejamos transformá-la em igualdade, o que temos a fazer é subtrair a variável não negativa w , isto é,

$$a_{l1}x_1 + a_{l2}x_2 + \cdots + a_{ln}x_n - w = b_l, \quad w \geq 0.$$

Por fim, se a l -ésima restrição for dada por uma igualdade, isto é,

$$a_{l1}x_1 + a_{l2}x_2 + \cdots + a_{ln}x_n = b_l,$$

então ela pode ser convertida em duas desigualdades:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i,$$

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i.$$

Agora, suponha que todas as restrições não canônicas se apresentam sob a forma de igualdade e, por algum motivo, uma das variáveis de decisão, digamos x_k , para algum $1 \leq k \leq n$, possa assumir valores também negativos. Afim de que o problema possua variáveis de decisão não negativas apenas, podemos proceder de duas maneiras:

- (i) identificar uma restrição onde o coeficiente de x_k é não nulo e cujas as demais variáveis de decisão envolvidas sejam todas não negativas. A seguir, escrever x_k como função das demais variáveis e substituir, nas demais restrições do problema, x_k pela expressão linear obtida. Sob tal escolha, o número de variáveis do problema original fica reduzido a $n - 1$ pois x_k não irá figurar de maneira explícita no problema e seu valor pode ser resgatado usando os valores ótimos das variáveis de decisão não negativas associadas a ela na expressão linear sintetizada para x_k .
- (ii) escrever x_k como a diferença entre duas outras variáveis não negativas u_k e v_k , isto é, $x_k = u_k - v_k$, com $u_k, v_k \geq 0$. Assim, novamente, a variável x_k deixa de figurar explicitamente no problema, passando a figurar em seu lugar a diferença $u_k - v_k$. Este procedimento aumenta o número de variáveis do problema em uma unidade, isto é, o problema passa a ser definido em \mathbb{R}^{n+1} . Por fim, o valor de x_k é calculado a partir dos valores de u_k e v_k associados a alguma solução ótima para o problema, após tal transformação. Note que x_k assume valor negativo se $u_k < v_k$ e não negativo caso contrário.

Assim, em algum sentido, em se tratando de problemas reais, não há um padrão sobre como as restrições se apresentam, além da exigência de que sejam lineares, é claro. Porém, adotada uma forma de representação assumida como padrão, ela deverá ser utilizada ao longo de toda formulação afim de evitar ambiguidades.

Para definir de maneira mais clara o que é um problema de programação linear (PPL), introduzimos sua formulação padrão, segundo [11], que consiste em:

$$\text{minimizar } z = \sum_{j=1}^n c_j x_j \tag{2.1}$$

sujeito a:

$$\sum_{j=1}^n a_{ij}x_j = b_i, i = 1, \dots, m, \quad (2.2)$$

$$x_j \geq 0, j = 1, 2, \dots, n, \quad (2.3)$$

onde c_j, a_{ij} e b_i são números reais dados e x_j representa, para $i = 1, 2, \dots, m; j = 1, 2, \dots, n$, as variáveis de decisão. A função linear a ser maximizada em (2.1) é denominada função objetivo. As restrições de não negatividade (2.3) são conhecidas como triviais. A mesma formulação pode ser expressa no formato matricial como:

$$(PPL): \text{ minimizar } z = c^T x \quad (2.4)$$

sujeito a:

$$Ax = b \quad (2.5)$$

$$x \geq 0, \quad (2.6)$$

onde $c^T = (c_1 \ c_2 \dots \ c_n)$, $x^T = (x_1 \ x_2 \ \dots \ x_n)$, $b^T = (b_1 \ b_2 \ \dots \ b_m)$, $A = [a_1 a_2 \dots a_n]$, onde $a_j \in \mathbb{R}^m$ denota a j -ésima coluna de A . A desigualdade 2.6 indica que cada componente do vetor x é não negativa.

Em geral, diante dos exemplos expostos, a modelagem de um PPL é construída da seguinte forma:

- identificação das variáveis de decisão;
- geração da função objetivo;
- geração das restrições não canônicas.
- acréscimo das restrições canônicas.

2.2 Soluções básicas viáveis

Considere o modelo primal de programação linear definido pelas relações (2.1)-(2.3). Discutiremos aqui uma maneira muito prática de se obter pontos para o subconjunto de R^n determinado por (2.2) e (2.3). Para tanto, assumiremos inicialmente que:

\mathcal{H}_1 $m < n$. Na prática, isso significa que o sistema linear $Ax = b$ possui mais variáveis do que equações.

\mathcal{H}_2 $\text{posto}(A) = m$. Todas as restrições são relevantes para o problema pois as linhas de A , que determinam os lados esquerdos das equações são linearmente independentes.

Como consequências das hipóteses \mathcal{H}_1 e \mathcal{H}_2 , segue que:

C_1 A possui m colunas linearmente independentes.

C_2 A admite uma submatriz não-singular de ordem m .

De certo modo, se tivéssemos $m \geq n$ então o $\text{posto}(A) \leq n$. Assim, para $\text{posto}(A) = n$ existiriam $m - n$ linhas de A que poderiam ser escritas como combinação linear das outras n linhas restantes. Na prática, isso indicaria que $m - n$ restrições de igualdade já seriam naturalmente atendidas se as n restrições restantes o fossem (caso o conjunto viável seja não-vazio). Dessa forma, $m - n$ restrições de igualdade poderiam ser removidas do PPL. Dando continuidade a esta mesma linha de raciocínio (se o conjunto viável for não-vazio), teríamos um total de n restrições de igualdade restantes, após a exclusão das $m - n$ restrições redundantes. Caso as n restrições de igualdade restantes fossem todas relevantes, isto é, as n linhas restantes de A fossem linearmente independentes, então o sistema passaria a ser quadrado de ordem n e teria uma única solução, o quê resumiria a determinação de solução para o PPL a simples determinação da única solução do sistema $Ax = b$. Isso justifica a adoção das hipóteses \mathcal{H}_1 e \mathcal{H}_2 .

Agora, note que

$$Ax = \begin{pmatrix} \sum_{j=1}^n a_{1j}x_j \\ \vdots \\ \sum_{j=1}^n a_{mj}x_j \end{pmatrix} = x_1 \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \end{pmatrix} + \cdots + x_n \begin{pmatrix} a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix}.$$

Dessa forma, se a_1, a_2, \dots, a_n são as colunas de A , então a restrição vetorial $Ax = b$ implica que

$$x_1 a_1 + x_2 a_2 + \cdots + x_n a_n = b, \quad (2.7)$$

i. e., para que o problema definido por (2.1)-(2.3) seja viável é necessário que b pertença ao espaço gerado pelas colunas de A .

Como as hipóteses \mathcal{H}_1 e \mathcal{H}_2 são assumidas, podemos extrair, das n colunas de A , um conjunto de m colunas linearmente independentes, formando assim uma base para \mathbb{R}^m . Admita que o conjunto $\{j_1, \dots, j_m\} \subset \{1, \dots, n\}$ contém os índices de um conjunto de m colunas linearmente independentes de A . Denote por $B \in \mathbb{R}^{m \times m}$ a matriz formada por essas colunas e por $N \in \mathbb{R}^{m \times n-m}$ a matriz formada pelas colunas restantes de A . Então, existe um único vetor $x_B \in \mathbb{R}^m$, tal que

$$x_{j_1} a_{j_1} + \cdots + x_{j_m} a_{j_m} = b, \quad (2.8)$$

onde a_{j_1}, \dots, a_{j_m} são as colunas de A que formam B . Dessa forma, as $n - m$ colunas restantes de A , que formam N , não figuram em (2.8), ou ainda, as coordenadas associadas a elas em (2.8) são todas nulas. Fazendo $x_N \in \mathbb{R}^{n-m}$ identicamente igual ao vetor nulo, segue que

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix},$$

com $x_N = \mathbf{0} \in \mathbb{R}^{n-m}$, é uma solução para sistema $Ax = b$ pois

$$\begin{aligned} Ax &= [B \quad N] \begin{pmatrix} x_B \\ x_N \end{pmatrix} = Bx_B + Nx_N = Bx_B + N\mathbf{0} = Bx_B \\ &= x_{j_1} a_{j_1} + \cdots + x_{j_m} a_{j_m} = b. \end{aligned}$$

Portanto,

$$x = \begin{pmatrix} x_B \\ \mathbf{0} \end{pmatrix},$$

é uma solução para o sistema linear $Ax = b$. Soluções para o sistema $Ax = b$ sob tal formato são denominadas soluções básicas, com respeito a uma base B pré-fixada. Segue que, para cada base escolhida, existe uma única solução básica associada a mesma.

Diante do exposto anteriormente e das definições já apresentadas, surge um questionamento: *e se não for possível identificar uma solução básica simplesmente olhando para as restrições não canônicas do problema de programação linear? Como proceder neste caso?*

Segundo (Maculam e Fampa, 2004) [23], quando não temos uma solução básica viável para o problema de programação linear, podemos acrescentar variáveis artificiais $g_i \geq 0$ à esquerda de cada restrição $\sum_{j=1}^n a_{ij}x_j = b_i$, $i = 1, 2, \dots, m$. De fato, suponha que $b_i \geq 0$, $i = 1, 2, \dots, m$. Considere o conjunto de restrições:

$$\sum_{j=1}^n a_{ij}x_j + g_i = b_i, i = 1, 2, \dots, m \quad (2.9)$$

$$x_j \geq 0, j = 1, 2, \dots, n \quad (2.10)$$

$$g_i \geq 0, i = 1, 2, \dots, m \quad (2.11)$$

Tomando como função objetivo a soma das variáveis artificiais adicionadas ao problema, temos:

$$(P) : \text{minimizar } \sum_{i=1}^m g_i$$

sujeito a

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j + g_i &= b_i, i = 1, 2, \dots, m \\ x_j &\geq 0, j = 1, 2, \dots, n \\ g_i &\geq 0, i = 1, 2, \dots, m \end{aligned}$$

É fácil verificar que as variáveis $x_j = 0, j = 1, 2, \dots, n$ e $g_i = b_i \geq 0$ estão associadas a uma solução básica de (2.9), (2.10) e (2.11).

Seja $v(P)$ o valor ótimo da função objetivo do problema de otimização linear artificial (P). Se $v(P) > 0$, temos que o conjunto de restrições do problema é vazio, e se $v(P) = 0$ a solução ótima obtida para (P) terá $g_i = 0, i = 1, 2, \dots, m$ e $x_j = \bar{x}_j \geq 0$, para $j = 1, 2, \dots, n$, satisfazendo $\sum_{j=1}^n a_{ij}\bar{x}_j = b_i, i = 1, 2, \dots, m$.

Se a base final da solução ótima de (P) não contiver nenhuma coluna associada às variáveis artificiais $g_i, i = 1, 2, \dots, m$, esta será também uma base primal viável do problema original. Caso a solução básica ótima encontrada para o (P) seja degenerada, sendo uma solução degenerada (em geral, soluções representadas por mais de uma base), há pelo menos uma coluna associada a g_i na base ótima e $g_i = 0$.

2.3 O Teorema Fundamental da Programação Linear

Nesta seção é destacado o Teorema Fundamental da Programação Linear, assim como sua demonstração. Esta versão diz respeito ao formato padrão, isto é, o objetivo baseado em minimização e as restrições não triviais escritas sob a forma de igualdade. O Teorema e sua demonstração foram extraídos de [11].

Teorema Fundamental da Programação Linear

Considere o PPL em seu formato vetorial padrão, onde $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, com $m < n$, $\text{posto}(A) = m$ e $b \in \mathbb{R}^m$. As seguintes afirmações são verdadeiras:

- (i) se o PL é viável, isto é, $V = \{x \in \mathbb{R}^n : Ax = b; x \geq \mathbf{0}\} \neq \emptyset$ então existe uma solução básica viável;

(ii) se existe uma solução (viável) ótima para o PL então existe uma solução básica (viável) ótima.

• **Demonstração**

(i). Seja $x \in V$ uma solução viável para o (PL). Denote por $a_j, j = 1, \dots, n$, a j -ésima coluna de A . Então,

$$x_1 a_1 + \dots + x_n a_n = b.$$

Assuma que $p \leq n$ componentes de x são não-nulas e que os índices dessas componentes são j_1, \dots, j_p . Dessa forma,

$$x_{j_1} a_{j_1} + \dots + x_{j_p} a_{j_p} = b. \quad (2.12)$$

A partir deste ponto, a demonstração pode ser dividida em dois casos:

(i.1) Admita que a_{j_1}, \dots, a_{j_p} são linearmente independentes. Neste caso, b é unicamente determinado pela combinação linear

$$x_{j_1} a_{j_1} + \dots + x_{j_p} a_{j_p} = b.$$

Como o número máximo de colunas linearmente independentes de A é m , pela hipótese \mathcal{H}_2 , segue que $p \leq m$. Portanto, temos dois subcasos do caso (i.1). São eles:

(i.1.1) se $p = m$ então construa a nova solução básica \tilde{x} fazendo $\tilde{x}_{j_l} = x_{j_l}, 1 \leq l \leq m$ e $\tilde{x}_{j_l} = 0, m+1 \leq l \leq n$ (solução básica viável não-degenerada).

(i.1.2) se $p < m$ então construa a nova solução básica $\tilde{x} \in \mathbb{R}^n \in \mathbb{R}^n$ fazendo $\tilde{x}_{j_l} = x_{j_l}, 1 \leq l \leq p$ e $\tilde{x}_{j_l} = 0, p+1 \leq l \leq n$ (solução básica viável degenerada, pois a variável básica $\tilde{x}_{j_l} = 0, \forall p+1 \leq l \leq m$).

(i.2) Admita agora que a_{j_1}, \dots, a_{j_p} são linearmente dependentes. Então, existem escalares $y_{j_l} \in \mathbb{R}, 1 \leq l \leq p$, não todos simultaneamente nulos, tais que

$$y_{j_1} A_{j_1} + y_{j_2} A_{j_2} + \dots + y_{j_p} A_{j_p} = \mathbf{0}.$$

Multiplicando ambos os membros da última igualdade por $-\delta$, para algum $\delta \in \mathbb{R}$, e adicionando a igualdade (2.12), segue que:

$$(x_{j_1} - \delta y_{j_1}) A_{j_1} + (x_{j_2} - \delta y_{j_2}) A_{j_2} + \dots + (x_{j_p} - \delta y_{j_p}) A_{j_p} = b. \quad (2.13)$$

A partir deste ponto, devemos nos perguntar acerca do limite para incrementar o parâmetro δ , a partir de 0, de maneira que as coordenadas $x_{j_l} - \delta y_{j_l}, l = 1, \dots, p$, possam compor uma solução básica viável, pois para $\delta = 0$, x já o é. Para tanto, consideraremos os seguintes casos:

($y_{j_l} \leq 0$) nesse caso, $-\delta y_{j_l} \geq 0$ e $x_{j_l} - \delta y_{j_l} \geq 0$ para qualquer escolha de $\delta \geq 0$, pois $x_{j_l} \geq 0, \forall 1 \leq l \leq p$.

($y_{j_l} > 0$) nesse caso, $-\delta y_{j_l} \leq 0$ e, para que $x_{j_l} - \delta y_{j_l} \geq 0$, devemos ter $\delta \leq \frac{x_{j_l}}{y_{j_l}}$.

De ambos os casos listados anteriormente, conclui-se que a escolha

$$\delta = \min\left\{\frac{x_{j_l}}{y_{j_l}} : y_{j_l} > 0\right\}, \quad (2.14)$$

garante que $x_{j_l} - \delta y_{j_l} \geq 0, \forall 1 \leq l \leq p$. Em particular, se $1 \leq q \leq p$ é o índice para o qual $y_{j_q} > 0$ e $\delta = \frac{x_{j_q}}{y_{j_q}}$ então $x_{j_q} - \delta y_{j_q} = 0$. Tendo a observação destacada, consideraremos agora os seguintes casos:

(i.2.1) se $p > m$ então incrementamos δ até o limite estabelecido em (2.14) e retiramos a coluna a_{j_q} , visto que a variável $x_{j_q} = 0$. Dessa forma, passamos a um conjunto formado por $p-1$ colunas satisfazendo a igualdade (2.13). O procedimento pode ser repetido até que se tenha apenas m variáveis não-nulas em (2.13). Caso as m variáveis não-nulas restantes estejam associadas a m colunas linearmente independentes de A então a nova solução básica $\tilde{x} \in \mathbb{R}^n$ será construída fazendo $\tilde{x}_{j_{l_k}} = x_{j_{l_k}} - \delta y_{j_{l_k}}$, onde $j_{l_k}, \forall 1 \leq k \leq m$, agora, são os índices referentes as m colunas restantes linearmente independentes de A em (2.13), e $\tilde{x}_{j_{l_k}} = 0, \forall m+1 \leq k \leq n$ (solução básica viável não-degenerada). Caso contrário, o procedimento adotado, a partir deste ponto, é o mesmo descrito no próximo caso.

(i.2.2) se $p = m$ então devemos também incrementar δ até o limite estabelecido em (2.14) e retirar a coluna A_{j_q} , visto que a variável $x_{j_q} = 0$, restando agora um conjunto com $m-1$ variáveis não-nulas em (2.13). Nesse momento, devemos verificar se as $m-1$ colunas restantes são linearmente independentes. Em caso afirmativo, dar-se-á o procedimento por encerrado e a nova solução básica viável $\tilde{x} \in \mathbb{R}^n$ será $\tilde{x}_{j_{l_k}} = x_{j_{l_k}} - \delta y_{j_{l_k}}, \forall 1 \leq k \leq m-1$ e $\tilde{x}_{j_{l_k}} = 0, \forall m \leq k \leq n$ (solução básica viável degenerada, com apenas uma variável básica nula). Caso contrário, o procedimento deve ser repetido até que se tenha um conjunto com $r < m$ variáveis não-nulas associadas a r colunas linearmente independentes. Assim sendo, a nova solução básica viável $\tilde{x} \in \mathbb{R}^n$ será $\tilde{x}_{j_{l_k}} = x_{j_{l_k}} - \delta y_{j_{l_k}}, \forall 1 \leq k \leq r$ e $\tilde{x}_{j_{l_k}} = 0, \forall r+1 \leq k \leq n$ (solução básica viável degenerada, com $m-r$ variáveis básicas nulas).

(ii) Suponha que $x^* \in \mathbb{R}^n$ é uma solução (viável) ótima para o PPL. Novamente, admita que x^* possui p coordenadas não nulas tais que

$$x_{j_1}^* a_{j_1} + \dots + x_{j_p}^* a_{j_p} = b. \quad (2.15)$$

Como x^* é viável para o PPL, podemos construir uma solução básica viável \tilde{x}^* , a partir de x^* , usando os passos descritos em (i). Portanto, podemos considerar também dois casos:

(ii.1) $\{a_{j_l} : 1 \leq l \leq p\}$ é um conjunto linearmente independente. Nesse caso, assim como em ambos os subcasos (i.1.1) e (i.1.2) de (i.1), a solução básica viável obtida coincide com a própria solução viável apresentada. Assim sendo, $\tilde{x}^* = x^*$. Logo, \tilde{x}^* também será ótima, além de básica viável, pois $c^T \tilde{x}^* = c^T x^*$.

(ii.2) $\{A_{j_l} : 1 \leq l \leq p\}$ é LD. Nesse caso, assim como em (i.2), a solução básica viável obtida \tilde{x}^* é da forma $\tilde{x}^* = x^* - \delta y$. Devemos mostrar então que \tilde{x}^* é também ótima. Em primeiro lugar, note que existe $\tilde{\delta}$ suficientemente pequeno em módulo para o qual $x^* - \tilde{\delta} y$ é viável, independente de seu sinal, i.e., independente se $\tilde{\delta} \leq 0$ ou $\tilde{\delta} \geq 0$. De fato, basta definir $\delta_1 = \min\{\frac{x_{j_l}^*}{y_{j_l}} : y_{j_l} > 0\}$ e $\delta_2 = \max\{\frac{x_{j_l}^*}{y_{j_l}} : y_{j_l} < 0\}$ e fazer $|\tilde{\delta}| = \min\{\delta_1, |\delta_2|\}$. Dessa forma, $\tilde{x}^* = x^* - \tilde{\delta} y$ é viável e

$$c^T \tilde{x}^* = c^T (x^* - \tilde{\delta} y) = c^T x^* - \tilde{\delta} c^T y.$$

Por outro lado, perceba que $c^T y = 0$, pois se assim não fosse, para $c^T y \neq 0$ poderíamos escolher $\bar{\delta}$, com $|\bar{\delta}| < |\tilde{\delta}|$ e sinal adequado (positivo ou negativo), tal que $\bar{x}^* = x^* - \bar{\delta} y$ é também viável e

$$c^T \bar{x}^* = c^T x^* - \bar{\delta} c^T y < c^T x^*,$$

o quê contradiria a otimalidade de x^* . Portanto, $c^T \tilde{x}^* = c^T x^*$, ou seja, \tilde{x}^* é solução básica viável ótima, o que completa a demonstração do teorema.

2.4 A geometria de um PPL

Nesta seção são abordados conceitos acerca da geometria de um PPL. É introduzido o conceito de convexidade para conjuntos, o conceito de ponto extremo ou vértice da região viável e, por fim, é demonstrada a equivalência existente entre os conceitos de solução básica e ponto extremo. Os conceitos e o resultado de equivalência entre soluções básicas e pontos extremos, assim como sua demonstração foram extraídos e adaptados de [11].

Um conjunto não-vazio $C \subseteq \mathbb{R}^n$ é dito convexo se para quaisquer $x^1, x^2 \in C$, o segmento de reta conectando x^1 à x^2 está inteiramente contido em C . Note que a equação vetorial da reta que contém os pontos x^1 e x^2 é da forma

$$\vec{O}x = \vec{O}x^1 + t(\vec{O}x^2 - \vec{O}x^1), \quad (2.16)$$

onde $\vec{O}x, \vec{O}x^1$ e $\vec{O}x^2$ representam os vetores com extremidades em x, x^1 e x^2 , respectivamente, e origem em $\mathbf{0} \in \mathbb{R}^n$. O parâmetro $t \in \mathbb{R}$ associa números reais à pontos da reta contendo x^1 e x^2 . à saber, $t = 0$ corresponde ao ponto x^1 e $t = 1$, ao ponto x^2 . Como as componentes dos vetores $\vec{O}x, \vec{O}x^1$ e $\vec{O}x^2$ correspondem as coordenadas dos pontos x, x^1 e x^2 , por abuso de notação, a equação (2.16) pode ser reescrita como

$$x(t) = x^1 + t(x^2 - x^1) = (1 - t)x^1 + tx^2, \quad t \in \mathbb{R}.$$

Os pontos associados aos valores de $t < 0$ e $t > 1$ correspondem a pontos da reta que se encontram à esquerda de x^1 e à direita de x^2 , respectivamente. Os pontos do segmento determinado por x^1 e x^2 correspondem aos valores de $t \in [0, 1]$.

Um ponto $x \in C \subset \mathbb{R}^n$ é dito *extremo* se **não** existem pontos $x^1, x^2 \in C$, tal que x pode ser escrito como

$$(1 - t)x^1 + tx^2, \quad (2.17)$$

para qualquer $0 < t < 1$.

A expressão (2.17) é denominada *combinação convexa* entre x^1 e x^2 . Em outras palavras, x é um ponto extremo de C se ele não pode ser escrito como uma combinação convexa de outros dois pontos de C .

Note que os pontos extremos de uma região viável, geometricamente, são as interseções entre as restrições não canônicas, bem como a interseção das mesmas com os semi-eixos positivos, isto é, são exatamente os vértices da região delimitada pelas restrições. Vale ressaltar que o conjunto formado pelos vértices da região viável para um PPL é um conjunto de cardinalidade finita.

Teorema (Soluções Básicas x Pontos Extremos)

Considere o PPL em seu formato padrão e satisfazendo \mathcal{H}_1 e \mathcal{H}_2 . Denote por $V = \{x \in \mathbb{R}^n : Ax = b; x \geq 0\}$ sua região viável. Então, x é um ponto extremo ou vértice de V se, e somente se é uma solução básica viável para o PPL.

• Demonstração

(\Rightarrow) Seja $x \in V$ uma solução básica, com respeito uma base $B \in \mathbb{R}^{m \times m}$, e a_{j_1}, \dots, a_{j_m} as colunas básicas de A que formam B . Como as coordenadas de x que estão associadas as colunas não-básicas são nulas, b é unicamente determinado por:

$$x_{j_1} a_{j_1} + \dots + x_{j_m} a_{j_m} = b. \quad (2.18)$$

Suponha, por contradição, que existam $\bar{x}, \tilde{x} \in V$, para os quais $x = (1 - t)\bar{x} + t\tilde{x}$, para algum $t \in (0, 1)$. Como $\bar{x}_j \geq 0, \tilde{x}_j \geq 0, 1 \leq j \leq n$ e $0 < t < 1$, $x_j = (1 - t)\bar{x}_j + t\tilde{x}_j \geq 0, 1 \leq j \leq n$,

com $x_{j_l} = \bar{x}_{j_l} = \tilde{x}_{j_l} = 0, m+1 \leq l \leq n$, pois x é solução básica (note que, para $m+1 \leq l \leq n$, $0 = x_{j_l} = (1-t)\bar{x}_{j_l} + t\tilde{x}_{j_l}$ e, nesse caso, $\bar{x}_{j_l} = \tilde{x}_{j_l} = 0$, pois o lado direito da igualdade é dado pela soma de duas parcelas não-negativas). Dessa forma, segue, do fato de $\bar{x}, \tilde{x} \in V$, que

$$\bar{x}_{j_1} a_{j_1} + \cdots + \bar{x}_{j_m} a_{j_m} = b \quad (2.19)$$

e

$$\tilde{x}_{j_1} a_{j_1} + \cdots + \tilde{x}_{j_m} a_{j_m} = b. \quad (2.20)$$

De (2.18), (2.19) e (2.20), conclui-se que $x_{j_l} = \bar{x}_{j_l} = \tilde{x}_{j_l}, 1 \leq l \leq m$, pois $\{A_{j_l} : 1 \leq l \leq m\}$ é linearmente independente (b é obtido de maneira única como combinação linear das colunas básicas a_{j_1}, \dots, a_{j_m} , i.e., se existem três combinações lineares dessas colunas resultando em b então as coordenadas correspondentes em cada uma delas devem ser iguais). Portanto, $x = \bar{x} = \tilde{x}$, ou ainda, x é ponto extremo de V pois não pode ser escrito como combinação convexa de outros dois pontos distintos de V .

(\Leftarrow) Seja x é um ponto extremo de V . Admita que x possui p componentes não-nulas, denotadas por x_{j_1}, \dots, x_{j_p} . Dessa forma,

$$x_{j_1} a_{j_1} + \cdots + x_{j_p} a_{j_p} = b, \quad (2.21)$$

com $x_{j_l} > 0, 1 \leq l \leq p$. Para mostrar que x é solução básica viável é preciso assegurar que a_{j_1}, \dots, a_{j_p} é LI. Ora, se assim não fosse, existiriam y_{j_1}, \dots, y_{j_p} , não todos nulos, tais que

$$y_{j_1} a_{j_1} + \cdots + y_{j_p} a_{j_p} = 0. \quad (2.22)$$

Multiplicando (2.22) por $t \in \mathbb{R}$ e subtraindo de (2.21), obtemos

$$(x_{j_1} - ty_{j_1})A_{j_1} + \cdots + (x_{j_p} - ty_{j_p})A_{j_p} = b. \quad (2.23)$$

Como a igualdade é válida para qualquer valor de t , para que todas as coordenadas da combinação linear do lado esquerdo da igualdade (2.23) permaneçam não-negativas, t deve ser incrementado até, no máximo, $\bar{\delta} = \min\{\frac{x_{j_l}}{y_{j_l}} : y_{j_l} > 0\}$. Por outro lado, t pode ser reduzido até, no mínimo, $\tilde{\delta} = \max\{\frac{x_{j_l}}{y_{j_l}} : y_{j_l} < 0\}$ para continuar garantindo a não-negatividade das mesmas coordenadas.

Dessa forma, a escolha de $\delta = \min\{\bar{\delta}, |\tilde{\delta}|\}$ implica na viabilidade de $\bar{x}, \tilde{x} \in \mathbb{R}^n$, definidos por $\bar{x}_{j_l} = x_{j_l} + \delta y_{j_l}$ e $\tilde{x}_{j_l} = x_{j_l} - \delta y_{j_l}, 1 \leq l \leq p$ e $\bar{x}_{j_l} = \tilde{x}_{j_l} = 0, p+1 \leq l \leq n$. Fazendo $y_{j_l} = 0, p+1 \leq l \leq n$, segue que $\bar{x} = x + \delta y$ e $\tilde{x} = x - \delta y$. Além disso,

$$\frac{1}{2}\bar{x} + \frac{1}{2}\tilde{x} = \frac{1}{2}(x + \delta y) + \frac{1}{2}(x - \delta y) = x,$$

o quê é uma contradição, pois x é ponto extremo e, conseqüentemente, não pode ser escrito como combinação convexa entre \bar{x} e \tilde{x} , uma vez que, por construção, $\bar{x}, \tilde{x} \in V$. Logo, $\{A_{j_l} : 1 \leq l \leq p\}$ é linearmente independente. Portanto, $p \leq m$ e o resultado segue.

2.5 Algoritmo Simplex primal

O método simplex é um método iterativo utilizado para se determinar, numericamente, a solução ótima de um modelo de Programação Linear.

Trata-se de uma metodologia que percorre os vértices da região viável, a saber, soluções básicas, em busca de um vértice ou solução básica viável ótima. Tal procedimento tem seus

alicerces estruturados sobre o Teorema Fundamental da Programação Linear, que assegura a existência de soluções básicas viáveis ótimas se o problema admite solução (viável) ótima. O algoritmo Simplex realiza na verdade três passos que são repetidos iterativamente até que a solução ótima seja alcançada. Eles aparecem descritos na sequência.

Dada uma solução básica viável, associada a uma base, o primeiro passo consiste em estabelecer um critério de otimalidade, isto é, um critério para a escolha de uma coluna (ou variável) para entrar na base de maneira a diminuir o valor da função objetivo.

De fato, seja $x \in \mathbb{R}^n$ uma solução viável para o PPL em seu formato padrão. Então,

$$z = c^T x = c_1 x_1 + \cdots + c_n x_n. \quad (2.24)$$

Se x é uma solução básica, cujas variáveis básicas são aquelas associadas aos índices em $\overline{\mathcal{J}} = \{j_1, \dots, j_m\}$, então a igualdade (2.24) fica reduzida à

$$z = c^T x = c_B^T x_B + c_N^T x_N = c_B^T x_B + c_B^T \mathbf{0} = c_B^T x_B = c_{j_1} x_{j_1} + \cdots + c_{j_m} x_{j_m} = z^{(0)}, \quad (2.25)$$

onde B representa a base formada pelas colunas de A cujos índices pertencem a $\overline{\mathcal{J}}$. Reordenando as colunas de A segundo os índices associados às variáveis básicas e as não-básicas, o tableu associado a solução básica x pode ser reescrito como

$$\begin{array}{cccccccc} x_{j_1} & x_{j_2} & \cdots & x_{j_m} & x_{j_{m+1}} & x_{j_{m+2}} & \cdots & x_{j_n} & b \\ 1 & 0 & \cdots & 0 & y_{1j_{m+1}} & y_{1j_{m+2}} & \cdots & y_{1j_n} & y_{10} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & y_{mj_{m+1}} & y_{mj_{m+2}} & \cdots & y_{mj_n} & y_{m0} \end{array}$$

Segue portanto do tableu que

$$x_{j_1} = y_{10} - \sum_{i=m+1}^n y_{1j_i} x_{j_i}, \quad (2.26)$$

\vdots

$$x_{j_m} = y_{m0} - \sum_{i=m+1}^n y_{mj_i} x_{j_i}. \quad (2.27)$$

Por outro lado, a substituição de (2.25) à (2.27) em (2.24) resulta em

$$\begin{aligned} Z &= c^T x = c_B^T x_B + c_N^T x_N = c_{j_1} x_{j_1} + \cdots + c_{j_m} x_{j_m} + c_{j_{m+1}} x_{j_{m+1}} + \cdots + c_{j_n} x_{j_n} \\ &= c_{j_1} (y_{10} - \sum_{i=m+1}^n y_{1j_i} x_{j_i}) + \cdots + c_{j_m} (y_{m0} - \sum_{i=m+1}^n y_{mj_i} x_{j_i}) + \\ &+ c_{j_{m+1}} x_{j_{m+1}} + \cdots + c_{j_n} x_{j_n} = c_{j_1} y_{10} + \cdots + c_{j_m} y_{m0} + \\ &+ [c_{j_{m+1}} - (c_{j_1} y_{1j_{m+1}} + \cdots + c_{j_m} y_{mj_{m+1}})] x_{j_{m+1}} + \\ &\vdots \\ &+ [c_{j_n} - (c_{j_1} y_{1j_n} + \cdots + c_{j_m} y_{mj_n})] x_{j_n} \\ &= c_{j_1} y_{10} + \cdots + c_{j_m} y_{m0} + (c_{j_{m+1}} - z_{j_{m+1}}) x_{j_{m+1}} + \cdots + (c_{j_n} - z_{j_n}) x_{j_n}, \end{aligned}$$

onde $z_{j_i} = \sum_{k=1}^m c_{j_k} y_{kj_i}$, $m+1 \leq i \leq n$. Note que

$$z_{j_i} = c_B^T a_{j_i}, \quad m+1 \leq i \leq n. \quad (2.28)$$

Lembrando que $x_{j_i} = y_{i0} = b_i$, $1 \leq i \leq m$ e a igualdade (2.25), vem que

$$z = c^T x = z^{(0)} + (c_{j_{m+1}} - z_{j_{m+1}}) x_{j_{m+1}} + \cdots + (c_{j_n} - z_{j_n}) x_{j_n}, \quad (2.29)$$

com z_{j_i} , $m+1 \leq i \leq n$, calculado como explicitado em (2.28).

Além disso, segue imediatamente de (2.29) que a entrada de uma coluna (ou uma variável) não-básica $a_{j_i} (x_{j_i})$, $m+1 \leq i \leq n$, na base só produz decréscimo na função objetivo para o PPL se o coeficiente $c_{j_i} - z_{j_i} < 0$, para algum $m+1 \leq i \leq n$. Portanto, tem-se:

- (o₁) Se, para algum $m+1 \leq i \leq n$, $c_{j_i} - z_{j_i} < 0$ então a coluna a_{j_i} pode ser elencada para entrar na base, ou seja, a variável x_{j_i} pode ser transformada em uma variável básica.
- (o₂) Se $c_{j_i} - z_{j_i} \geq 0, \forall i = m+1, \dots, n$, então x é uma solução básica ótima para o PPL. Essa condição é conhecida como critério de otimalidade primal.

Suponha que $x \in \mathbb{R}^n$ é uma solução básica não-degenerada e não-ótima, associada a uma base formada pelas colunas cujos índices pertencem ao conjunto $\overline{\mathcal{J}} = \{j_1, \dots, j_m\}$. Agora, abordaremos um critério para escolha da coluna a deixar a base, tendo em vista que elencamos uma coluna de A , que estava fora da base, para entrar na base. Note que

$$x_{j_1} a_{j_1} + \dots + x_{j_m} a_{j_m} = b. \quad (2.30)$$

Admita que se deseja trazer a coluna não-básica a_{j_q} , para algum $m+1 \leq q \leq n$ para base. Além disso, existem $y_{j_1 j_q}, \dots, y_{j_m j_q} \in \mathbb{R}$, tal que

$$a_{j_q} = y_{j_1 j_q} a_{j_1} + \dots + y_{j_m j_q} a_{j_m}, \quad (2.31)$$

pois a_{j_1}, \dots, a_{j_m} são linearmente independentes. Multiplicando (2.31) por $\delta \geq 0$ e subtraindo o resultado de (2.30), segue que:

$$\left(x_{j_1} - \delta y_{j_1 j_q}\right) A_{j_1} + \left(x_{j_2} - \delta y_{j_2 j_q}\right) A_{j_2} + \dots + \left(x_{j_m} - \delta y_{j_m j_q}\right) A_{j_m} + \delta A_{j_q} = b. \quad (2.32)$$

Analisando (2.32), temos que

- (i) para $y_{j_i j_q} < 0$, para algum $1 \leq i \leq m$, δ pode ser incrementado sem qualquer limitação superior, pois $x_{j_i} - \delta y_{j_i j_q} > 0, \forall \delta \geq 0$.
- (ii) para $y_{j_i j_q} = 0$, para algum $1 \leq i \leq m$, incrementos em δ não influenciam em $x_{j_i} - \delta y_{j_i j_q} = x_{j_i} > 0$
- (iii) para $y_{j_i j_q} > 0$, para algum $1 \leq i \leq m$, δ deve ser incrementado até que $x_{j_p} - \delta y_{j_p j_q} = 0$, onde $1 \leq p \leq m$ é o primeiro índice para o qual isso ocorre. Ainda nesse caso, conclui-se que o valor máximo que δ pode assumir é

$$\delta = \min \left\{ \frac{x_{j_i}}{y_{j_i j_q}} : y_{j_i j_q} > 0 \right\} = \frac{x_{j_p}}{y_{j_p j_q}}, \quad (2.33)$$

e, assim sendo, a coluna A_{j_p} passa a não mais figurar em (2.32), dando lugar a coluna A_{j_q} . A maneira de se calcular δ , em (2.33), é denominada teste da razão.

Dessa forma, seja $x \in \mathbb{R}^n$ uma solução básica viável para o PPL, com respeito a uma base B , formada pelas colunas de A com índice em $\overline{\mathcal{J}} = \{j_1, \dots, j_m\}$. Denote por $N \in \mathbb{R}^{m \times n-m}$ a matriz formada pelas colunas de A com índice em $\mathcal{J} - \overline{\mathcal{J}}$, onde $\mathcal{J} = \{1, \dots, n\}$. O Algoritmo Simplex pode ser resumido pelos passos descritos a seguir.

• Algoritmo Simplex Primal

- P_1 . Calcule $c_N - z \in \mathbb{R}^{n-m}$, onde $z_{j_i} = c_B^T a_{j_i}, m+1 \leq i \leq n$. Se $c_N - z \geq 0$ então pare o algoritmo e retorne x como solução ótima, caso contrário, escolha um índice $j_q \in \mathcal{J} - \overline{\mathcal{J}}$, para o qual $c_{j_i} - z_{j_i} < 0$ e elenque $A_{j_q} \left(x_{j_q}\right)$ para entrar na base.

P_2 . Calcule $\delta = \min \left\{ \frac{x_{ji}}{y_{jjq}} : y_{jjq} > 0 \right\}$. Seja $j_p \in \overline{\mathcal{J}}$ o índice para o qual $\delta = \frac{x_{j_p}}{y_{j_p j_q}}$. Então escolha a coluna $a_{j_p}(x_{j_p})$ para sair da base.

P_3 . Efetue a mudança de base, onde $a_{j_q}(x_{j_q})$ entra na base e $A_{j_p}(x_{j_p})$ sai da base, atualize $\overline{\mathcal{J}}$ e retorne ao passo P_1 .

Vale ressaltar que a mudança de base indicada em P_3 pode ser facilmente obtida realizando-se operações elementares sobre as linhas da matriz ampliada $[A|b]$ do sistema linear $Ax = b$, ou seja, sobre as linhas do interior do tableu associado ao problema, como é mostrado no exemplo que segue.

Exemplo 1: considere o seguinte modelo de PPL :

$$\begin{aligned} \max \quad & 3x_1 + x_2 + 3x_3 \\ \text{s.a} \quad & x_1 + 2x_2 + 3x_3 \leq 5 \\ & 2x_1 + 2x_2 + x_3 \leq 6 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \end{aligned}$$

Convertendo para o formato padrão, obtemos:

$$\begin{aligned} \min. \quad & -3x_1 \quad -x_2 \quad -3x_3 \quad +0 \cdot x_4 \quad +0 \cdot x_5 \quad +0 \cdot x_6 \\ \text{s. a} \quad & 2x_1 \quad +x_2 \quad +x_3 \quad +x_4 \quad \quad \quad \quad \quad = 2 \\ & x_1 \quad +2x_2 \quad +3x_3 \quad \quad \quad +x_5 \quad \quad \quad = 5 \\ & 2x_1 \quad +2x_2 \quad +x_3 \quad \quad \quad \quad \quad +x_6 \quad = 6 \\ & x_j \geq 0, \quad 1 \leq i \leq 6. \end{aligned}$$

Calculando z_j e já utilizando o resultado para compor o tableu, segue que:

$$\begin{aligned} z_1 &= c_B^T A_1 = 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 2 = 0 \\ z_2 &= c_B^T A_2 = 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 2 = 0 \\ z_3 &= c_B^T A_3 = 0 \cdot 1 + 0 \cdot 3 + 0 \cdot 1 = 0. \end{aligned}$$

c	-3	-1	-3	0	0	0	b
r_1	2	1	1	1	0	0	2
r_2	1	2	3	0	1	0	5
r_3	2	2	1	0	0	1	6
$c_j - z_j$	-3	-1	-3	0	0	0	-

A última linha mostra que qualquer uma das colunas A_1, A_2 ou A_3 é candidata a entrar na base. Seguindo uma perspectiva "gulosa", escolheremos $A_1(x_1)$ para entrar na base, pois seu custo é um dos maiores em módulo. Outras perspectivas podem ser adotadas, como por exemplo uma escolha que reduza o número de divisões, evitando assim instabilidade numérica do método. Calculando δ , segue que $\delta = \min \left\{ \frac{2}{2}, \frac{5}{1}, \frac{6}{2} \right\} = \min \{1, 3, 5\} = 1$. Como o mínimo ocorre para a razão dada pelo valor da 1ª variável básica, que nesse caso é x_4 , a coluna a deixar a base é a 4ª. Assim sendo, devemos fazer, consecutivamente, $L_1 \leftrightarrow \frac{1}{2}L_1, L_2 \leftrightarrow L_2 + (-1)L_1$ e $L_3 \leftrightarrow L_3 + (-2)L_1$. Dessa forma, o tableu é atualizado como

c	-3	-1	-3	0	0	0	b
r_1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	1
r_2	0	$\frac{3}{2}$	$\frac{5}{2}$	$-\frac{1}{2}$	1	0	4
r_3	0	1	0	-1	0	1	4
$c_j - z_j$	0	$\frac{1}{2}$	$-\frac{3}{2}$	$\frac{3}{2}$	0	0	-

pois,

$$z_2 = c_B^T A_2 = -3 \cdot \frac{1}{2} + 0 \cdot \frac{3}{2} + 0 \cdot 1 = -\frac{3}{2},$$

$$z_3 = c_B^T A_3 = -3 \cdot \frac{1}{2} + 0 \cdot \frac{5}{2} + 0 \cdot 0 = -\frac{3}{2},$$

$$z_4 = c_B^T A_4 = -3 \cdot \frac{1}{2} + 0 \cdot \left(-\frac{1}{2}\right) + 0 \cdot (-1) = -\frac{3}{2}.$$

Note que apenas $A_3(x_3)$ é candidata a entrar na base. Novamente, calculando δ , segue que $\delta = \min \left\{ \frac{1}{\frac{1}{2}}, \frac{4}{\frac{5}{2}} \right\} = \min \left\{ 2, \frac{8}{5} \right\} = \frac{8}{5}$. Como o mínimo agora ocorre para a razão dada pelo valor da 2ª variável básica, que nesse caso é x_5 , a coluna a deixar a base é a 5ª. Assim sendo, devemos fazer, consecutivamente, $L_2 \leftrightarrow \frac{2}{5}L_2$

$$L_1 \leftrightarrow L_1 + \left(-\frac{1}{2}\right)L_2$$

Dessa forma, o tableu é reescrito como

c	-3	-1	-3	0	0	0	b
r_1	1	$\frac{1}{5}$	0	$\frac{3}{5}$	$-\frac{1}{5}$	0	$\frac{1}{5}$
r_2	0	$\frac{3}{5}$	1	$-\frac{1}{5}$	$\frac{2}{5}$	0	$\frac{8}{5}$
r_3	0	1	0	-1	0	1	4
$c_j - z_j$	0	$\frac{7}{5}$	0	$\frac{6}{5}$	$\frac{3}{5}$	0	-

pois,

$$z_2 = c_B^T A_2 = -3 \cdot \frac{1}{5} - 3 \cdot \frac{3}{5} + 0 \cdot 1 = -\frac{12}{5},$$

$$z_4 = c_B^T A_4 = -3 \cdot \frac{3}{5} - 3 \cdot \left(-\frac{1}{5}\right) + 0 \cdot (-1) = -\frac{6}{5},$$

$$z_5 = c_B^T A_5 = -3 \cdot \left(-\frac{1}{5}\right) - 3 \cdot \left(\frac{2}{5}\right) + 0 \cdot 0 = -\frac{3}{5}.$$

Como na última linha do tableu $c_j - z_j \geq 0, 1 \leq j \leq 6$, este último quadro é ótimo. Portanto, a solução ótima para o modelo inicial é $X_1^* = \frac{1}{5}, X_2^* = 0$ e $x_3^* = \frac{8}{5}$, com valor ótimo dado por $z^* = -\left(-3 \cdot \frac{1}{5} - 1 \cdot 0 - 3 \cdot \frac{8}{5}\right) = \frac{3}{5} + \frac{24}{5} = \frac{27}{5}$. Além disso, as duas primeiras restrições são atendidas sem folga ($x_4^* = X_5^* = 0$) e a última, com uma folga igual à 4 ($x_6^* = 4$).

O modelo dual de programação linear

3.1 Definição e exemplos

A teoria de dualidade em otimização linear é um componente fundamental e essencial para compreender a natureza dos problemas. Quando se trata de problemas de maximização ou minimização sujeitos a restrições lineares, o conceito de dualidade oferece uma perspectiva complementar do problema original. Nesta seção, os conceitos e exemplos abordados também foram extraídos e adaptados de [10], [11] e [12].

Seja (P) o problema de programação linear dado por,

$$\begin{aligned} &\text{minimizar} && \sum_{j=1}^n c_j x_j \\ &\text{sujeito a} && \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, m, \\ &&& x_j \geq 0, \quad j = 1, \dots, n. \end{aligned}$$

onde $c_j^s, a_{ij}^s, b_i^s \in \mathbb{R}$.

Denote por $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$ o vetor contendo os custos, a matriz dos coeficientes das restrições e o vetor contendo os termos independentes, respectivamente. (P) pode ser reescrito no formato vetorial como

$$\begin{aligned} &\text{minimizar} && c^T x \\ &\text{sujeito a} && Ax \geq b, \\ &&& x \geq 0, \\ &&& x \in \mathbb{R}^n. \end{aligned}$$

O problema dual (D) do problema (P) é definido como

$$\begin{aligned} &\text{maximizar} && \sum_{i=1}^m b_i y_i \\ &\text{sujeito a} && \sum_{i=1}^m a_{ij} y_i \leq c_j, \quad j = 1, \dots, n \\ &&& y_i \geq 0, \quad i = 1, \dots, m. \end{aligned}$$

O problema (D) pode ser reescrito no formato vetorial como

$$\begin{aligned} &\text{maximizar} && b^T y \\ &\text{sujeito a} && A^T y \leq c, \\ &&& y \geq 0, \quad y \in \mathbb{R}^m. \end{aligned}$$

Exemplo 2: considere o problema de programação linear em seu formato padrão

$$\begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito a} & Ax = b, \\ & x \geq 0. \end{array}$$

A restrição vetorial de igualdade pode ser convertida em duas restrições vetoriais de desigualdade

$$Ax = b \Leftrightarrow Ax \geq b \text{ e } Ax \leq b.$$

Multiplicando ambos os membros da 2ª desigualdade por (-1), o PL pode ainda ser reescrito como

$$\begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito a} & Ax \geq b \\ & -Ax \geq -b \\ & x \geq 0 \end{array}$$

ou ainda, como

$$\begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito a} & \begin{bmatrix} A \\ -A \end{bmatrix} x \geq \begin{bmatrix} b \\ -b \end{bmatrix}, \\ & x \geq 0 \end{array}$$

Dessa forma, o dual correspondente é dado por.

$$\begin{array}{ll} \text{maximizar} & b^T u - b^T v \\ \text{sujeito a} & A^T u - A^T v \leq c, \\ & u \geq 0, v \geq 0, u, v \in \mathbb{R}^m. \end{array}$$

ou ainda, por

$$\begin{array}{ll} \text{maximizar} & b^T (u - v) \\ \text{sujeito a} & A^T (u - v) \leq c, \\ & u \geq 0, v \geq 0, u, v \in \mathbb{R}^m. \end{array}$$

Fazendo $y = u - v$, o problema dual pode ser finalmente reescrito como

$$\begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito a} & A^T y \leq c, \\ & y \in \mathbb{R}^m. \end{array}$$

Vale observar que se (P) apresenta-se no formato padrão então as variáveis duais são livres (não obedecem restrições canônicas). De uma maneira geral, as variáveis duais associadas as restrições de igualdade no primal são livres.

Exemplo 3: considere agora o problema

$$\begin{array}{ll} \text{minimizar} & -x_1 - 4x_2 - 3x_3 \\ \text{sujeito a} & 2x_1 + 2x_2 + x_3 \leq 4, \\ & 2x_1 + 2x_2 + x_3 \leq 6, \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array}$$

O mesmo pode ser reescrito como

$$\begin{array}{ll} \text{minimizar} & -x_1 - 4x_2 - 3x_3 \\ \text{sujeito a} & -2x_1 - 2x_2 - x_3 \geq -4 \\ & -2x_1 - 2x_2 - x_3 \geq -6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{array}$$

Seu dual é dado por

$$\begin{array}{ll} \text{maximizar} & -4y_1 - 6y_2 \\ \text{sujeito a} & -2y_1 - 2y_2 \leq -1 \\ & -2y_1 - 2y_2 \leq -4 \\ & -y_1 - y_2 \leq -3, \quad y_1 \geq 0, y_2 \geq 0. \end{array}$$

Fazendo $\bar{y}_1 = -y_1$ e $\bar{y}_2 = -y_2$, o dual é reescrito como

$$\begin{array}{ll} \text{maximizar} & 4\bar{y}_1 - 6\bar{y}_2 \\ \text{sujeito a} & 2\bar{y}_1 + 2\bar{y}_2 \leq -1 \\ & 2\bar{y}_1 + 2\bar{y}_2 \leq -4 \\ & \bar{y}_1 + \bar{y}_2 \leq -3, \\ & \bar{y}_1 \leq 0, \bar{y}_2 \leq 0. \end{array}$$

3.2 Aspectos gerais da teoria da dualidade

A dualidade na programação linear está intimamente ligada à existência de um problema dual para cada problema primal. O Teorema da Dualidade estabelece que, se uma solução ótima existe para o problema primal, então uma solução ótima também existe para o problema dual, e ambos possuem o mesmo valor ótimo.

Os problemas primal e dual estão interligados por meio de correspondências entre variáveis e restrições. As variáveis do problema dual correspondem às restrições do problema primal e o contrário também é válido.

Além de oferecer certificados de otimalidade, a teoria da dualidade é fundamental na interpretação econômica das soluções ótimas, fornecendo informações sobre custos marginais, preços sombra e valores de recursos. Ela também é útil na análise de sensibilidade, permitindo entender como mudanças nos parâmetros afetam as soluções ótimas.

Teorema de Dualidade Fraca

Sejam (P) o problema primal de otimização linear em seu formato padrão e (D) o dual correspondente, isto é,

$$\begin{array}{ll} \text{minimizar} & c^T x \\ (P) \text{ sujeito a} & Ax = b, \\ & x \geq 0. \end{array} \quad \begin{array}{ll} \text{maximizar} & b^T y \\ (D) \text{ sujeito a} & A^T y \leq c, \\ & y \in \mathbb{R}^m. \end{array}$$

Sejam $x \in \mathbb{R}^n$ e $y \in \mathbb{R}^m$ pontos viáveis para (P) e (D) , respectivamente. Então,

$$c^T x \geq b^T y.$$

• Demonstração

De fato, se $(x, y) \in \mathbb{R}^n \times \mathbb{R}^m$ é um par primal-dual viável então

$$b^T y = (Ax)^T y = x^T A^T y \leq x^T c = c^T x.$$

A desigualdade segue do fato de $x \geq 0$ e $A^T y \leq c$.

Teorema de Dualidade Forte

Sejam (P) o problema primal de otimização linear em seu formato padrão e (D) , seu dual correspondente, isto é,

$$\begin{array}{ll} \text{minimizar} & c^T x \\ (P) \text{ sujeito a} & Ax = b, \\ & x \geq 0. \end{array} \quad \begin{array}{ll} \text{maximizar} & b^T y \\ (D) \text{ sujeito a} & A^T y \leq c. \\ & y \in \mathbb{R}^m. \end{array}$$

- (i) Se um dos problemas, ou (P) ou (D) , possui solução ótima com valor objetivo ótimo não finito então o outro é inviável.
- (ii) Se um dos problemas, ou (P) ou (D) , possui solução ótima com valor objetivo ótimo finito então o outro também possui e o valor objetivo ótimo de ambos os problemas coincidem.

• Demonstração

De fato, (i) segue como corolário imediato do Teorema de Dualidade Fraca. Agora, para (ii), suponha que (P) possui uma solução ótima x^* com valor objetivo ótimo $z^* = c^T x^*$ finito. Defina o seguinte conjunto

$$S = \{r = tz^* - c^T x, w = tb - Ax, x \geq 0, t \geq 0; x \in \mathbb{R}^n, t \in \mathbb{R}\}.$$

Segue que S é um cone (pois $\alpha(r, w) \in S$, para todo $\alpha \geq 0$ e $(r, w) \in S$) convexo e fechado. Note que $(1, 0) \notin S$. Com efeito, se $w = \bar{t}b - A\bar{x} = \vec{0}$, para algum $\bar{t} > 0$ e $\bar{x} \geq 0$, então $x = \bar{x}/\bar{t}$ é viável para (P) e portanto $r/\bar{t} = z^* - c^T x \leq 0$, o que implica em $r \leq 0$. Se $w = -A\bar{x} = \vec{0}$, para algum $\bar{x} \geq 0$ e $c^T \bar{x} = -1$ e, ainda, se $x \in \mathbb{R}^n$ é viável para (P) então $x + \alpha \bar{x}$ também o é, $\forall \alpha \geq 0$.

Por outro lado, $c^T(x + \alpha \bar{x}) = c^T x + \alpha c^T \bar{x} = c^T x + \alpha \cdot (-1) = c^T x - \alpha$. Dessa forma, para α suficientemente grande, o valor objetivo primal não pode ser limitado inferiormente, o que contradiz o fato do valor objetivo ótimo de (P) ser finito. Isto implica que não pode existir \bar{x} satisfzendo tais condições. Logo, a afirmação que $(1, \vec{0}) \notin C$ é verdadeira. Pelo Teorema de existência de hiperplano de separação entre um conjunto convexo e um ponto que não pertence a seu fecho, existem $(s, y) \in \mathbb{R} \times \mathbb{R}^m$, não-nulo, tal que

$$\inf_{(r,w) \in S} sr + w^T y > s$$

Se existir $(r, w) \in S$ tal que $sr + w^T y < 0$ então $\alpha(sr + w^T y) \rightarrow -\infty$, quando $\alpha \rightarrow +\infty$. Mas, $\alpha(r, w) \in S$, para todo $\alpha \geq 0$. Isto viola o Teorema de existência de hiperplano de separação entre um conjunto convexo e um ponto que não pertence a seu fecho, pois s não assume valor finito nesse caso. Logo, $\inf_{(r,w) \in S} sr + w^T y \geq 0$. Por outro lado, como

$$(0, \vec{0}) \in S$$

$$\inf_{(r,w) \in S} sr + w^T y \leq s \cdot 0 + \vec{0}^T \lambda = 0$$

Portanto, $\inf_{(r,w) \in S} sr + w^T y = 0$ e, como consequência, $s < 0$. Sem perda de generalidade, assumamos que $s = -1$. Então, existe $\lambda \in \mathbb{R}^m$ tal que

$$-r + w^T \lambda \geq 0,$$

para todo $(r, w) \in S$. Aplicando a definição de S , segue que

$$x^T (c - A^T \lambda) - tz^* + tb^T y \geq 0$$

para todo $x \geq 0, t \geq 0$ (note que $c^T x = x^T c$). Fazendo $t = 0$, segue que $A^T y \leq c$, i.e., y é dual viável. Fazendo ainda $x = 0$ e $t = 1$, segue que $b^T y \geq z^*$. Mas, pelo corolário c do teorema de dualidade fraca, $b^T y \leq z^*$. Logo, $b^T y = z^*$, o que conclui a demonstração.

Vale enfatizar que alguns pontos da demonstração do Teorema de Dualidade Forte são assumidos aqui sem argumentação prévia. Dentre eles, destacam-se a convexidade e o fechamento topológico do conjunto S , assim como o Teorema de existência de hiperplanos de separação entre conjuntos convexos fechados. Contudo, a convexidade e o fechamento topológico de S podem ser verificados de maneira direta, sem a necessidade de introdução de novos resultados. Já, o Teorema de existência de hiperplanos de separação entre conjuntos convexos fechados pode ser encontrado em [10].

Segue também que o dual (D) de um problema de otimização linear em seu formato padrão, também é um *PPL* cujas variáveis são livres. Além disso, após análise de alguns aspectos do problema dual (variáveis livres, desigualdades, maximização), o simplex pode ser facilmente adaptado para resolver também (D) .

Corolário (Dualidade Forte)

Sejam (P) o problema primal de otimização linear em seu formato padrão e (D) , seu dual correspondente, isto é,

$$\begin{array}{ll} \text{minimizar} & c^T x \\ (P) \text{ sujeito a} & Ax = b, \\ & x \geq 0. \end{array} \quad \begin{array}{ll} \text{maximizar} & b^T y \\ (D) \text{ sujeito a} & A^T y \leq c. \\ & y \in \mathbb{R}^m. \end{array}$$

Se (P) possui uma solução básica ótima associada a uma base B então $y = B^{-1T} c_B$ é uma solução ótima para (D) . Além disso, os valores objetivos ótimos de ambos os problemas coincidem.

• Demonstração

Sejam $x \in \mathbb{R}^n$ uma solução básica (viável) ótima para (P) e $A = [B \mid N]$ a partição ótima de A em colunas associada a x , com $B \in \mathbb{R}^{m \times m}$ não-singular. Como $x = (x_B \mid x_N)$, com $x_B = B^{-1} b$ e $x_N = \vec{0}$ é ótima, $r_N = c_N - N^T B^{-1T} c_B$, cujas componentes negativas indicariam as variáveis não-básicas candidatas a entrarem na base, possui todas as componentes não-negativas. Isso implica que $N^T B^{-1T} c_B \leq c_N$. Defina $y = B^{-1T} c_B$. Segue que

$$A^T y = [B^T y \mid N^T y] = [c_B \mid N^T B^{-1T} c_B] \leq [c_B \mid c_N] = c.$$

Portanto, $A^T y \leq c$, i.e., y é dual viável. Por outro lado

$$b^T y = b^T (B^{-1T} c_B) = (b^T B^{-1T} c_B)^T = c_B^T B^{-1} b = c_B^T x_B,$$

que é o valor objetivo ótimo para o problema primal. Logo, y é uma solução ótima dual, o que completa a demonstração.

Exemplo 4: considere o problema primal (P)

$$\begin{array}{ll} \text{minimizar} & -x_1 - 4x_2 - 3x_3 \\ \text{Sujeito a} & 2x_1 + 2x_2 + x_3 \leq 4 \\ & x_1 + 2x_2 + 2x_3 \leq 6 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0. \end{array}$$

Após a adição das variáveis de folga, o tableaux associado a P , reescrito no formato padrão, é

$$\begin{array}{c|cccccc|c} r_1 & 2 & 2 & 1 & 1 & 0 & 4 \\ r_2 & 1 & 2 & 2 & 0 & 1 & 6 \\ \hline c_j - z_j & -1 & -4 & -3 & 0 & 0 & 0 \end{array}$$

onde r_1 e r_2 representam os coeficientes da 1^a e 2^a restrição, respectivamente, c_j a componente de c associada a j -ésima variável não-básica e z_j o produto interno entre as componentes de c associadas as variáveis básicas e a j -ésima coluna não-básica do tableaux. Os tableaux's que seguem com a aplicação do simplex são

$$\begin{array}{c|cccccc|c} r_1 & 1 & 1 & 1/2 & 1/2 & 0 & 2 \\ r_2 & -1 & 0 & 1 & -1 & 1 & 2 \\ \hline c_j - z_j & 3 & 0 & -1 & 2 & 0 & 8 \end{array}$$

e, finalmente,

$$\begin{array}{c|cccccc|c} r_1 & 3/2 & 1 & 0 & 1 & -1/2 & 1 \\ r_2 & -1 & 0 & 1 & -1 & 1 & 2 \\ \hline c_j - z_j & 2 & 0 & 0 & 1 & 1 & 10. \end{array}$$

A solução ótima para (P) é $x_1 = 0, x_2 = 1$ e $x_3 = 2$, com valor ótimo -10 . Agora, multiplicando-se as restrições de (P) por -1 , como segue

$$\begin{array}{ll} \text{minimizar} & -x_1 - 4x_2 - 3x_3 \\ \text{Sujeito a} & -2x_1 - 2x_2 - x_3 \geq -4 \\ & -x_1 - 2x_2 - 2x_3 \geq -6 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \end{array}$$

o dual (D) é dado por

$$\begin{array}{ll} \text{maximizar} & -4y_1 - 6y_2 \\ \text{sujeito a} & -2y_1 - y_2 \leq -1, \\ & -2y_1 - 2y_2 \leq -4, \\ & -y_1 - 2y_2 \leq -3, \\ & y_1 \geq 0, y_2 \geq 0. \end{array}$$

O valor ótimo de (D) é atingido em $y_1 = 1, y_2 = 1$ e é também igual a -10 , como mostra a Figura (3.1).

3.3 Algoritmo Simplex dual

A seguir, apresentamos o esquema associado ao Algoritmo Simplex dual. A formulação detalhada do mesmo pode ser encontrada em [11].

- **Algoritmo Simplex Dual**

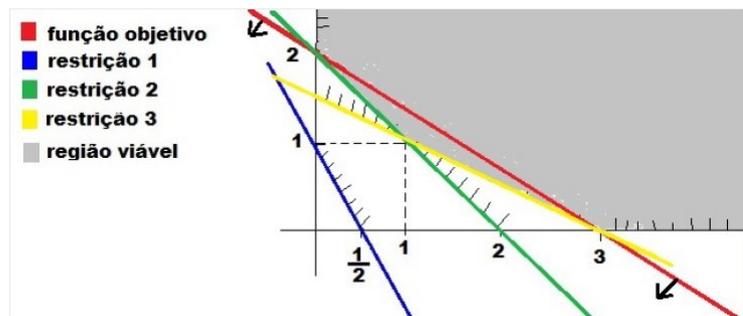


Figura 3.1 – Gráfico do valor ótimo

P_1 . Dada uma solução dual viável x_B , se $x_B \geq 0$ então x_B é primal ótima e $y = B^{-1T} c_B$ é a solução dual ótima correspondente. Caso contrário, selecione alguma componente $x_{Bi} < 0$ de x_B .

P_2 . Se $y_{ij} \geq 0, j = 1, \dots, n$, então o valor ótimo do dual é infinito e, portanto, (P) é inviável. Se $y_{il} < 0$ para algum $1 \leq l \leq n$, então defina

$$\epsilon_0 = \frac{z_k - c_k}{y_{ik}} = \min_j \left\{ \frac{z_j - c_j}{y_{ij}} : y_{ij} < 0 \right\}.$$

P_3 . Construa uma nova base \tilde{B} , substituindo a_i por a_k . De posse \tilde{B} , calcule a solução básica viável dual correspondente $x_{\tilde{B}}$ e retorne ao passo P_1 .

Exemplo 5: considere o problema

$$\begin{aligned} \text{Minimizar} \quad & 3x_1 + 4x_2 + 5x_3 \\ \text{Sujeito a} \quad & x_1 + 2x_2 + 3x_3 \geq 5 \\ & 2x_1 + 2x_2 + x_3 \geq 6 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

Multiplicando as desigualdades por -1 e introduzindo variáveis de folga obtem-se o seguinte tableau

c	3	4	5	0	0	
r_1	-1	-2	-3	1	0	-5
r_2	-2	-2	-1	0	1	-6
$c_j - z_j$	3	4	5	0	0	

A base corresponde a uma solução dual viável, pois $c_j - z_j \geq 0$. O próximo passo é Selecionar algum $x_{Bi} < 0$, por exemplo $x_5 = -6$, para sair da base. Note que $\epsilon_0 = \min\{-3/-2, -4/-2, -5/-1\} = 3/2$. Dessa forma, a coluna a substituir a coluna básica a_5 é a coluna a_1 . A partir daí, procedemos a mudança de base.

c	3	4	5	0	0	
r_1	0	-1	-5/2	1	-1/2	-2
r_2	1	1	1/2	0	-1/2	3
$c_j - z_j$	0	1	7/2	0	3/2	

Agora $x_4 = -2$ é a única escolha para sair da base. Além disso, $\epsilon_0 = \min\{-1/-1, -7/2 \div -5/2, -3/2 \div -1/2\} = 1$. Dessa forma, a coluna a substituir a coluna básica a_4 é a coluna a_2 .

Procedendo novamente a mudança de base, temos o tableau

c	3	4	5	0	0	
r_1	0	1	5/2	-1	1/2	2
r_2	1	0	-2	1	-1	1
$c_j - z_j$	0	0	1	1	1	

O último tableau produz uma solução viável para o primal que, portanto, é ótima. A saber, $x_1 = 1, x_2 = 2, x_3 = 0$.

3.4 Algoritmo simplex primal-dual

Na presente seção, também optamos por apenas sintetizar o Algoritmo Simplex primal-dual, sem abordar detalhes de sua formulação, tendo em vista que nosso objetivo é explicitar as técnicas e metodologias que implicitamente figuram na síntese de problemas de otimização linear inteira. De maneira análoga à ressaltada na seção anterior, a formulação detalhada do algoritmo Simplex primal-dual pode ser encontrada também em [11].

Novamente, considere o *PPL* em seu formato primal padrão:

$$(P) \quad \begin{array}{ll} \text{minimizar} & c^T x \\ \text{sujeito} & Ax = b \\ & x \geq 0 \end{array}$$

e o seu dual correspondente

$$(D) \quad \begin{array}{ll} \text{maximizar} & b^T y \\ \text{sujeito} & A^T y \leq c. \end{array}$$

Em se tratando de restrições de desigualdade, um dos pontos mais importantes em otimização para o estabelecimento de condições de otimalidade, seja no caso linear, seja no não-linear, é a introdução do conceito de restrição ativa. As condições de KKT estabelecem que neste caso apenas as restrições ativas devem ser levada em consideração na síntese de condições de otimalidade para um ponto. Em outras palavras, sob a presença de restrições de desigualdade, um ponto será considerado um candidato a solução ótima se satisfaz as condições de Lagrange pois apenas as restrições ativas assumem papel determinante em sua caracterização.

Seja $y \in \mathbb{R}^m$ uma solução dual viável, isto é, $A^T y \leq c$. A i -ésima restrição $a_i^T y \leq c_i$ é dita ativa em y se $a_i^T y = c_i$, onde a_i é a i -ésima coluna de A .

O resgate do conceito de restrição ativa se dá claramente com base no modelo dual, associado ao modelo primal de otimização linear, tendo em vista que o mesmo contém apenas restrições de desigualdade em sua formulação.

Agora, denote por $\mathcal{S}(y) \subset \{1, \dots, n\}$ o conjunto de índices para os quais $a_i^T y = c_i$. Decorre imediatamente do conceito de restrição ativa que se $i \notin \mathcal{S}(y)$ então $a_i^T y < c_i$.

Admitindo ainda que $y \in \mathbb{R}^m$ uma solução dual viável, defina o seguinte problema primal restrito associado (*PPRA*):

$$(PPRA) \quad \begin{array}{ll} \text{minimizar} & e^T w \\ \text{sujeito} & Ax + w = b \\ & x \geq 0 \text{ e } x_i = 0, \text{ para } i \notin \mathcal{S}(y) \\ & y \geq 0, \end{array}$$

onde $e \in \mathbb{R}^m$ é tal que $e = (1, 1, \dots, 1)^T$. O problema dual restrito (PDRA), correspondente ao (PPRA), é dado por

$$\begin{aligned} & \text{maximizar} && u^T b \\ (PDRA) & \text{sujeito} && a_i^T u \leq 0, \quad i \notin \mathcal{I}(y) \\ & && u \leq 1. \end{aligned}$$

As condições de otimalidade para o algoritmo Simplex primal-dual podem ser expressas de acordo com o seguinte teorema:

Teorema

Suponha que \bar{y} é dual viável e que o par $(\bar{x}, w) \in \mathbb{R}^{n \times m}$, com $w = \mathbf{0}$, é viável para o problema primal restrito associado a \bar{y} . Então (\bar{x}, \bar{y}) é um par primal-dual ótimo para o problema primal original e seu dual correspondente.

• Demonstração

Note que se $w = \mathbf{0}$ então \bar{x} é viável para o primal. Além disso, $c^T \bar{x} = (A^T \bar{y})^T \bar{x}$, pois $A^T \bar{y} = c$ para as componentes de c correspondentes as componentes não-nulas de \bar{x} (note que, para $i \in \mathcal{I}(\bar{y})$, $a_i^T \bar{x} = c_i$ e, para $i \notin \mathcal{I}(\bar{y})$, $a_i^T \bar{x} = 0$, pois $\bar{x}_i = 0$). Logo, $c^T \bar{x} = (A^T \bar{y})^T \bar{x} = \bar{y}^T (A\bar{x}) = \bar{y}^T b = b^T \bar{y}$ e, pelo Teorema de Dualidade Fraca, o resultado segue.

É importante enfatizar que o método primal-dual inicia com uma solução viável para o dual e, então, otimiza o problema primal restrito associado. Se a solução do problema primal restrito associado é inviável para o primal então a solução dual é melhorada e um novo problema primal restrito associado é construído.

• Algoritmo Simplex Primal-Dual

- P_1 . Dada uma solução y_0 dual viável, faça $k = 0$.
- P_2 . Escreva o problema primal restrito associado a solução y_k .
- P_3 . Determine uma solução para problema primal restrito associado. Se o valor ótimo do problema é igual a zero então a solução obtida é ótima para o problema primal original, pelo último teorema.
- P_4 . Se o valor ótimo do problema primal restrito associado é positivo então obtenha do tableux simplex final para o problema primal restrito associado a solução u_k para seu dual (dual do problema primal restrito associado). Se $u_k^T a_j \leq 0$, para todo $j = 1, \dots, n$, então o primal é inviável.
- P_5 . Por outro lado, se $u_k^T a_l > 0$, para algum $1 \leq l \leq n$, então defina um novo vetor dual viável

$$y_{k+1} = y_k + \epsilon_k u_k,$$

onde $\epsilon_k = \min_j \{(c_j - y_k^T a_j) / u_k^T a_j : u_k^T a_j > 0\}$, faça $k = k + 1$ e retorne ao passo P_2 .

Exemplo 6: considere o seguinte PL em seu formato padrão

$$\begin{array}{ll}
\text{minimizar} & 2x_1 + x_2 + 4x_3 \\
\text{sujeto} & x_1 + x_2 + 2x_3 = 3 \\
& 2x_1 + x_2 + 3x_3 = 5 \\
& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.
\end{array}$$

Para este caso, introduzimos variáveis artificiais x_4 e x_5 ao problema, isto é,

$$\begin{array}{llllll}
\text{minimizar} & 2x_1 & +x_2 & +4x_3 & & \\
\text{sujeto} & x_1 & +x_2 & +2x_3 & +x_4 & = 3 \\
& 2x_1 & +x_2 & +3x_3 & & +x_5 = 5 \\
x_1 \geq 0, & x_2 \geq 0, & x_3 \geq 0, & x_4 \geq 0, & x_5 \geq 0.
\end{array}$$

Como os coeficientes da função objetivo são não-negativos, $y = (0, 0)$ é dual viável pois $c_i - z_i \geq 0$, para $i = 1, 2, 3$, onde $z_i = c_B^T a_i$. Dessa forma, iniciamos o tableau simplex primal-dual como

$$\begin{array}{rcccccc}
& & a_1 & a_2 & a_3 & \cdot & \cdot & b \\
r_1 & \rightarrow & 1 & 1 & 2 & 1 & 0 & 3 \\
r_2 & \rightarrow & 2 & 1 & 3 & 0 & 1 & 5 \\
\tilde{c}_i - \tilde{z}_i & \rightarrow & -3 & -2 & -5 & 0 & 0 & -8 \\
c_i - a_i^T y & \rightarrow & 2 & 1 & 4 & \cdot & \cdot & \cdot
\end{array}$$

onde \tilde{c} é o vetor de custos do problema primal restrito associado e $\tilde{z} = \tilde{c}_B^T a_i$.

As colunas a_i admissíveis para entrar na base são as relacionadas as restrições ativas em y_0 . Essa relação é encontrada para as colunas associadas aos elementos nulos da última linha ($a_i^T y_0 = c_i \Leftrightarrow c_i - a_i^T y_0 = 0$). No exemplo, como não há elementos nulos na última linha, não há restrições duais ativas em $y_0 = (0, 0)$. Portanto, não é possível reduzir o valor ótimo do problema primal restrito associado e a solução ótima associada à y_0 , é $x_1 = x_2 = x_3 = 0, x_4 = 3$ e $x_5 = 5$. A solução ótima dual para o dual do problema primal restrito associado é $u_0 = B^{-1T} \tilde{c}_B = l\tilde{c}_B = (1, 1)$ e $a_i^T u_0, i = 1, 2, 3$, são os valores simétricos dos 3 primeiros elementos na 3ª linha do tableau.

Dessa forma, deve-se computar $\epsilon_0 = \min_i \{\frac{2}{3}, \frac{1}{2}, \frac{4}{5}\} = \frac{1}{2}$. Isso produz $y_1 = y_0 + \epsilon_0 \cdot u_0 = (0, 0) + \frac{1}{2} \cdot (1, 1) = (\frac{1}{2}, \frac{1}{2})$ e o novo tableau associado a y_1 é dado por

$$\begin{array}{rcccccc}
& & a_1 & a_2 & a_3 & \cdot & \cdot & b \\
r_1 & \rightarrow & 1 & 1 & 2 & 1 & 0 & 3 \\
r_2 & \rightarrow & 2 & 1 & 3 & 0 & 1 & 5 \\
\tilde{c}_i - \tilde{z}_i & \rightarrow & -3 & -2 & -5 & 0 & 0 & -8 \\
c_i - a_i^T y_1 & \rightarrow & \frac{1}{2} & 0 & \frac{3}{2} & \cdot & \cdot & \cdot
\end{array}$$

Agora, tem-se o tableau associado a solução y_1 . Nesse caso, a variável x_2 é candidata a entrar na base, pois $a_2^T y_1 = c_2$.

Como $\delta_{\min} = \{\frac{3}{1}, \frac{5}{1}\} = 3, x_4$ sai da base. Segue daí que

$$\begin{array}{rcccccc}
& & a_1 & a_2 & a_3 & \cdot & \cdot & b \\
r_1 & \rightarrow & 1 & 1 & 2 & 1 & 0 & 3 \\
r_2 & \rightarrow & 1 & 0 & 1 & -1 & 1 & 2 \\
\tilde{c}_i - \tilde{z}_i & \rightarrow & -1 & 0 & -1 & 2 & 0 & -2 \\
c_i - a_i^T y_1 & \rightarrow & \frac{1}{2} & 0 & \frac{3}{2} & \cdot & \cdot & \cdot
\end{array}$$

Como não há mais colunas candidatas a entrar na base nessa etapa, a solução ótima para o novo problema primal restrito associado é $x_1 = x_3 = x_4 = 0, x_2 = 3$ e $x_5 = 2$. A solução ótima dual para o dual do problema primal restrito associado é $u_1 = B^{-1} \tilde{c}_B = I \tilde{c}_B = (0, 1)$ e $a_i^T u_1, i = 1, 2, 3$ são os valores simétricos dos 3 primeiros elementos na 3ª linha do tableau.

Dessa forma, deve-se computar $\epsilon_1 = \min_i \left\{ \frac{1}{2}, \frac{3}{1} \right\} = \frac{1}{2}$. Isso produz $y_2 = y_1 + \epsilon_1 \cdot u_1 = \left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2} \cdot (0, 1) = \left(\frac{1}{2}, 1\right)$ e o novo tableau associado a y_2 é dado por

$$\begin{array}{rcccccc}
 & & a_1 & a_2 & a_3 & \cdot & \cdot & b \\
 r_1 & \rightarrow & 1 & 1 & 2 & 1 & 0 & 3 \\
 r_2 & \rightarrow & 1 & 0 & 1 & -1 & 1 & 2 \\
 \tilde{c}_i - \tilde{z}_i & \rightarrow & -1 & 0 & -1 & 2 & 0 & -2 \\
 c_i - a_i^T y_1 & \rightarrow & 0 & 0 & 1 & \cdot & \cdot & \cdot
 \end{array}$$

Agora, tem-se o tableau associado a solução y_2 . Nesse caso, a variável x_1 é candidata a entrar na base, pois $a_1^T y_2 = c_1$.

Tableux simplex primal-dual (exemplo) Como $\delta_{\min} = \left\{ \frac{3}{1}, \frac{2}{1} \right\} = 2, x_5$ sai da base. Segue daí que

$$\begin{array}{rcccccc}
 & & a_1 & a_2 & a_3 & \cdot & \cdot & b \\
 r_1 & \rightarrow & 0 & 1 & 1 & 2 & -1 & 1 \\
 r_2 & \rightarrow & 1 & 0 & 1 & -1 & 1 & 2 \\
 \tilde{c}_i - \tilde{z}_i & \rightarrow & 0 & 0 & 0 & 1 & 1 & 0 \\
 c_i - a_i^T y_1 & \rightarrow & 0 & 0 & 1 & \cdot & \cdot & \cdot
 \end{array}$$

Como a solução primal é viável, segue que ela é também ótima, isto é, $x_1^* = 2, x_2^* = 1$ e $x_3^* = 0$.

Programação Linear Inteira

4.1 Programação Linear Inteira

4.1.1 Definição e aspectos gerais de otimização linear inteira

Os problemas de otimização linear inteira mantêm a mesma estrutura dos problemas de otimização linear convencionais. No entanto, é crucial ressaltar uma característica distintiva: as restrições aplicadas às variáveis exigem que todas sejam expressas apenas como números inteiros. Isso implica que apenas soluções viáveis que contêm variáveis com valores inteiros são consideradas aceitáveis para essa categoria de problema. A programação inteira pode ser categorizada como pura, onde apenas variáveis inteiras são utilizadas, ou mista, onde uma combinação de variáveis inteiras e fracionárias é empregada. Além disso, um problema de programação linear inteira pode ser binário, conforme definido por [24], no qual as variáveis de decisão têm valores restritos a 0 ou 1. Esta seção abordará os conceitos fundamentais da programação linear inteira que estão diretamente ligados aos objetivos da pesquisa.

Exemplo 7: para seguir a abordagem delineada por [23], o problema subsequente, extraído do mesmo texto, será considerado inicialmente.

$$\begin{array}{ll}
 \text{Maximizar} & z = x_1 - 3x_2 - 4x_3 \\
 & 2x_1 + x_2 - x_3 \leq 4 \\
 & 4x_1 - 3x_2 \leq 2 \\
 & 3x_1 + 2x_2 + x_3 \leq 3 \\
 \text{sujeito a} & x_1 \geq 0 \\
 & x_2 \geq 0 \\
 & x_3 \geq 0 \\
 & x_2, x_3 \in \mathbb{Z}.
 \end{array}$$

Segundo [23], o problema restringe x_2 e x_3 a valores inteiros não negativos, enquanto que x_1 pode assumir qualquer valor real não negativo.

Imaginemos a solução ignorando as restrições de integralidade. Para isso, o problema será considerado como sendo de programação linear. Caso a solução obtida forneça valores inteiros para todas as variáveis de decisão com restrição de integridade, temos o problema original também resolvido. Porém, se alguma variável com restrições de integridade assume valor fracionário na solução do problema linear, quando deveria ser inteira, instintivamente tenderíamos a arredondar o valor para o inteiro mais próximo de maneira a garantir ainda a viabilidade

imposta pelas restrições do problema. Infelizmente este procedimento fornece soluções inteiras distantes do ótimo. Para melhor compreensão, considere um outro exemplo destacado em [23]:

$$\begin{aligned}
 (P) : \text{ maximizar } & z = x_1 + 19x_2 \\
 & x_1 + 20x_2 \leq 50 \\
 \text{sujeito a } & x_1 + x_2 \leq 20 \\
 & x_1 \geq 0 \\
 & x_1, x_2 \in \mathbb{Z}.
 \end{aligned}$$

Relaxando as restrições de integralidade das variáveis x_2 e x_3 , o problema (P) se torna:

$$\begin{aligned}
 (\bar{P}) : \text{ maximizar } & z = x_1 + 19x_2 \\
 & x_1 + 20x_2 \leq 50 \\
 \text{sujeito a } & x_1 + x_2 \leq 20 \\
 & x_1 \geq 0 \\
 & x_2 \geq 0.
 \end{aligned}$$

Como (\bar{P}) é um problema de otimização linear, ele pode ser solucionado através do método simplex. O método Simplex é um procedimento iterativo que pode ser executado manualmente, para problemas de pequeno porte embora, para problemas de médio e grande porte, seja mais conveniente a utilização de pacotes computacionais consolidados no mercado, alguns deles com licença gratuita para pesquisas acadêmicas, ou a própria implementação do algoritmo.

No exemplo explicitado, aplicando o Simplex ao problema (\bar{P}) , a despeito de resolvê-lo graficamente, tendo em vista que se trata de um problema com apenas duas variáveis, segue que a solução ótima obtida é dada por $x_1^* = 18.421052631579$, $x_2^* = 1.5789473684211$, com valor ótimo $z^* = 48.421052631579$.

Sabendo que o valor ótimo para (P) não deve ser superior a $z^* = 48.421052631579$, tendo em vista que seus pontos viáveis, além de possuir apenas coordenadas inteiras, devem pertencer a região viável do problema relaxado (\bar{P}) , o arredondamento dos valores de x_1^* e x_2^* para 18 e 2, respectivamente, produziria um ponto inviável para (P) pois a primeira restrição não canônica deixaria de ser atendida. Por outro lado, o truncamento dos valores de x_1^* e x_2^* para 18 e 1, respectivamente, resultaria em um valor objetivo igual a 37. Porém, o valor ótimo de (P) , usando uma técnica bastante disseminada para resolução de problemas inteiros, conhecida como *branch and bound*, é igual à 48 e este valor está associado ao ponto (10, 2). Isso mostra que ambos os procedimentos, arredondamento ou truncamento de valores fracionários para variáveis inteiras, não levam ao ótimo do problema inteiro. Mais especificamente neste caso, o arredondamento pode inclusive resultar em pontos inviáveis para o problema.

4.2 Modelos lineares inteiros binários e mistos

A otimização inteira constitui uma área fundamental na otimização, dedicada a problemas nos quais as variáveis de decisão devem ser números inteiros. Dentro dos diversos tipos de modelos lineares inteiros, dois se destacam: binários, onde as variáveis só podem assumir valores binários (0 ou 1), e os mistos, que combinam variáveis inteiras e contínuas.

4.2.1 Modelos binários

Os modelos inteiros binários são uma ferramenta essencial na modelagem matemática, empregados em uma ampla gama de aplicações. Esses modelos representam situações onde as variáveis de decisão são restritas a assumir valores binários 0 ou 1. Eles desempenham um papel crucial em problemas que envolvem decisões discretas, como na alocação de recursos, planejamento de rotas, design de redes, entre outros. A natureza binária desses modelos permite representar de forma eficaz escolhas mútuas excludentes ou decisões 'sim/não', agregando simplicidade e poder na solução de problemas complexos.

Em [2], é destacado um exemplo clássico de um modelo linear inteiro conhecido como *problema da mochila*. Também conhecido como knapsack problem, o problema da mochila se destaca por sua estreita relação com diversos problemas de otimização. Em linhas gerais, o problema consiste em maximizar o valor dos itens carregados em uma mochila, sem ultrapassar o limite máximo de peso suportado por ela. Segundo [3], Dantzing possivelmente foi o primeiro pesquisador a documentá-lo na literatura, e desde então, o problema tem sido um marco nas técnicas de programação inteira, otimização combinatória e programação dinâmica. Além de sua importância matemática, o modelo em si tem aplicações diretas em situações práticas.

O problema da mochila pode ser formulado da seguinte forma:

$$(PM) \text{ Maximizar } z = \sum_{j=1}^n c_j x_j$$

sujeito a:

$$\sum_{j=1}^n w_j x_j \leq b$$

$$x_j \geq 0 \text{ e inteiro.}$$

onde x_j representa o número de objetos do tipo j selecionados para serem incluídos em uma mochila, representada pela restrição do modelo. A mochila possui uma capacidade total de b unidades. A variável c_j representa o valor econômico de cada item, e w_j , o peso do item. Na literatura correlata, utiliza-se, sem qualquer perda de generalidade, os coeficientes das restrições também inteiros. Nesse caso:

$$\sum_{j=1}^n w_j > b \text{ e } w_j \leq b, j = 1, \dots, n$$

Considerando que exista apenas um objeto de cada tipo para ser escolhido, define-se o problema da mochila 0-1, em que a restrição da variável inteira é substituída por $x_j \in \{0, 1\}$. Denominaremos essa formulação de *(PMI)*. O problema assim definido é também muitas vezes denominado de problema da mochila unidimensional (por possuir apenas uma restrição tipo mochila).

$$(PMI) \text{ Maximizar } z = \sum_{j=1}^n c_j x_j$$

sujeito a:

$$\sum_{j=1}^n w_j x_j \leq b$$

$$x_j \in \{0, 1\} \quad j = 1, \dots, n$$

Um caso particular bastante conhecido para o problema *(PMI)* é aquele em que as

variáveis de decisão são inteiras e limitadas em certos valores máximos. Esse problema é denominado da mochila, com limites, (*PML*) e pode ser formulado do seguinte modo:

$$(PML) \text{ Maximizar } z = \sum_{j=1}^n c_j x_j$$

sujeito a:

$$\begin{aligned} \sum_{j=1}^n w_j x_j &\leq b \\ x_j &\leq l_j \quad j = 1, \dots, n \\ x_j &\in \mathbb{S}^+ \end{aligned}$$

Os limites impostos pelos valores l_j não permitem que a mochila seja preenchida com uma quantidade qualquer de objetos x_j .

O (PM) é NP-difícil (ver Garey e Johnson [1979]). Já o problema da mochila linear, ou seja, aquele em que as variáveis são contínuas, pode ser solucionado de modo extremamente eficiente, em $O(n)$ (complexidade linear), onde n representa o número de variáveis do problema, podendo uma solução ser computada tanto usando Simplex, quanto através do método de pontos interiores, introduzido por Kamarkar, aplicado a resolução de problemas lineares.

O modelo da Mochila talvez seja um dos problemas de otimização em que fica mais evidente a diferença entre determinar uma solução quando as variáveis não são inteiras e depois que passam a assumir a restrição de integralidade.

4.2.2 Modelos inteiros mistos

Os modelos lineares inteiros mistos constituem uma classe crucial de problemas de otimização que envolvem variáveis inteiras e contínuas. Essa categoria de modelos permite a representação de situações do mundo real onde decisões discretas e contínuas estão interligadas. Suas variáveis podem ser restritas a assumir valores inteiros ou contínuos, oferecendo uma representação mais precisa de uma ampla gama de problemas complexos. Essa abordagem híbrida encontra aplicabilidade em diversos campos, desde logística e planejamento de produção até finanças e engenharia (Hillier and Lieberman, 2001) [25].

Por exemplo, no planejamento de rotas, pode-se utilizar esses modelos para determinar as melhores rotas para entrega, considerando restrições como capacidade de carga dos veículos e horários de entrega. Além disso, na alocação de recursos em empresas, os modelos inteiros mistos são usados para otimizar a alocação de mão-de-obra e máquinas para maximizar a produção.

Um exemplo clássico é o problema de *corte de estoque* [25]. Ele estabelece que uma empresa produz rolos de papel de diferentes larguras, e o objetivo é determinar quantos rolos de cada largura devem ser cortados de rolos-mãe, de largura fixa, de modo a satisfazer a demanda por rolos de diferentes larguras e, ao mesmo tempo, minimizar o desperdício de material.

Suponha que a empresa produza rolos-mãe de largura W polegadas e precise atender a pedidos de rolos de larguras l_i polegadas, $i = 1, \dots, m$. A empresa quer minimizar a quantidade total de rolos-mãe usados no processo de corte para atender à demanda. A formulação deste problema como um problema de programação linear inteira mista requer a decisão de determinar a quantidade de rolos de cada largura a serem cortados dos rolos-mãe, levando em consideração as restrições de largura e buscando minimizar o desperdício de material. As variáveis inteiras são utilizadas para definir o número de rolos cortados de cada rolo-mãe, a fim de

atender à demanda. Denote por x_{ij} a quantidade de j -ésimo rolo que será usada para atender a demanda de largura l_i e por d_i , a quantidade demandada de rolos de largura l_i . A formulação do problema é a seguinte:

$$\text{Minimizar } \sum_j x_{ij} \quad (\text{Minimizar o número total de rolos-mãe usados})$$

sujeito a:

$$\sum_i l_i \cdot x_{ij} \leq W \quad \text{para todo rolo-mãe } j$$

$$\sum_j x_{ij} = d_i \quad \text{para todo tipo de rolo demandado } i$$

$$x_{ij} \geq 0 \quad \text{e inteiro} \quad \text{para todo } i, j$$

O problema pode ser resolvido formulando-o com variáveis inteiras que representam as quantidades de padrões de corte e, em seguida, empregando algum *método de enumeração implícita ou de planos de corte* [5] para encontrar a solução ótima.

4.3 Método de enumeração e planos de corte

4.3.1 Métodos de enumeração - aspectos gerais

Em [26] vários métodos de otimização combinatória são abordados, incluindo o método de enumeração. Especificamente, este método é uma abordagem exaustiva que consiste em listar todas as possíveis soluções viáveis para um problema e, em seguida, seleciona a melhor delas com base em algum critério de otimalidade. No contexto de otimização combinatória, o método de enumeração é frequentemente utilizado para encontrar a solução ótima ou uma boa aproximação para problemas de pequeno a médio portes, uma vez que a busca exaustiva pode se tornar computacionalmente inviável para problemas de grande porte devido à sua complexidade exponencial.

Um exemplo clássico que utiliza o método de enumeração é o problema do caixeiro-viajante, visto em [2] no qual um viajante precisa visitar um conjunto de cidades uma única vez, voltando ao ponto de origem, e o objetivo é encontrar o caminho mais curto possível. Para o problema do caixeiro-viajante de pequeno porte, o método de enumeração pode ser aplicado para encontrar todas as permutações possíveis das cidades e calcular o comprimento de cada rota afim de determinar a mais curta. No entanto, à medida que o número de cidades aumenta, o número de permutações a serem avaliadas cresce exponencialmente, tornando essa abordagem impraticável para problemas de grande porte.

De acordo com [2], o problema do caixeiro-viajante pode ser formulado da seguinte forma:

$$\text{Minimizar } z = \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij}$$

sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N$$

Onde a variável binária x_{ij} assume valor igual a 1, se o arco $(i, j) \in A$ for escolhido para integrar a solução, e 0, caso contrário, e S é um subgrafo de G , em que $|S|$ representa sua cardinalidade. Nessa formulação assumimos implicitamente que x_{ii} não existe, que teremos $n(n-1)$ variáveis inteiras 0-1 e que o número total de restrições é de ordem $O(2^n)$. O conjunto de restrições determina a eliminação desses circuitos pré-hamiltonianos (caminhos que visitam todos os vértices de um grafo exatamente uma vez, exceto pelo vértice de partida, que é visitado duas vezes).

As equações em $|S|$ tornam os circuitos pré-hamiltonianos ilegais. A Figura mostra que a restrição para $|S| = 5$ elimina circuitos pré-hamiltonianos com cinco vértices da seguinte forma:

$$x_{15} + x_{54} + x_{49} + x_{96} + x_{61} \leq 4$$

$$x_{23} + x_{310} + x_{108} + x_{87} + x_{72} \leq 4$$

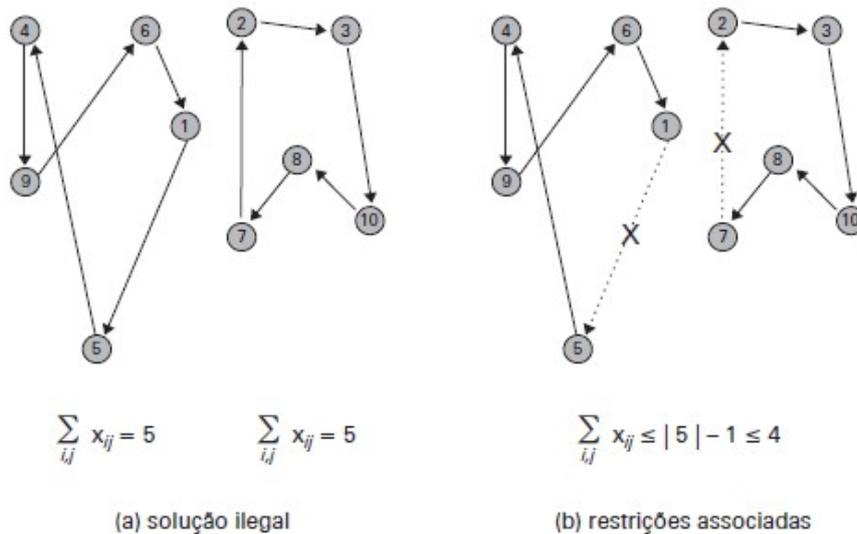


Figura 4.1 – Restrições de cardinalidade para o problema do caixeiro viajante. Extraído de [2], em 09/12/2023.

Para cada circuito pré-hamiltoniano possível, é necessária uma restrição do tipo $\sum_{i,j \in S} x_{ij} \leq |S| - 1$, $\forall S \subset N$, justificando-se assim um total de restrições da ordem $O(2^n)$.

Essa formulação destaca um importante aspecto do problema do caixeiro viajante, que é sua natureza combinatória. Pela formulação fica claro que solucionar um problema com essa estrutura é determinar uma certa permutação legal de custo mínimo. A formulação apresentada em [2] também auxilia no entendimento da ligação desse problema aos problemas de sequenciamento de operações, tão comuns em manufatura.

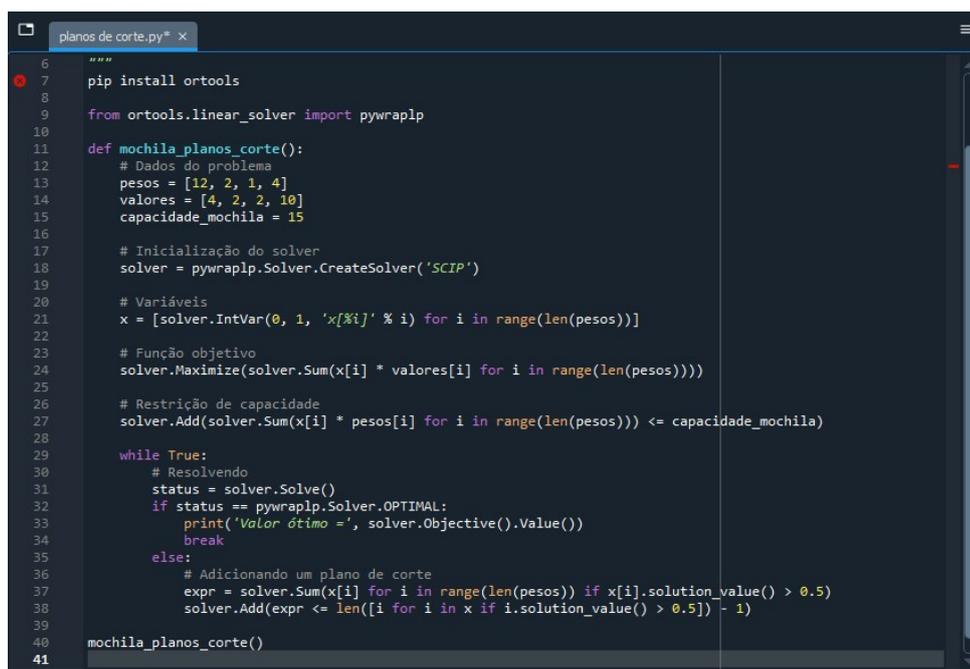
4.3.2 Planos de corte - aspectos gerais

Os planos de corte são usados em métodos de otimização [12], como o algoritmo Branch and Bound, para reduzir o espaço de busca. Eles são restrições adicionais que podem ser acrescentadas ao problema para eliminar ou reduzir o número de soluções a serem consideradas. Os cortes são normalmente desigualdades lineares que eliminam parte do espaço de busca, tornando-o menor e, portanto, mais fácil de explorar. Eles ajudam a evitar que o método de enumeração explore áreas do espaço de solução que podem não conter uma solução ideal.

Um exemplo clássico, visto em [25] seria o Problema da Mochila. A seguir, é apresentado um exemplo simples usando a técnica de planos de corte para resolver o Problema da Mochila:

Exemplo 7: Suponha que temos uma mochila com capacidade máxima de 15 unidades de peso e os seguintes itens:

- Item 1: Peso = 12, Valor = 4
- Item 2: Peso = 2, Valor = 2
- Item 3: Peso = 1, Valor = 2
- Item 4: Peso = 4, Valor = 10



```
planos de corte.py x
6 """
7 pip install ortools
8
9 from ortools.linear_solver import pywraplp
10
11 def mochila_planos_corte():
12     # Dados do problema
13     pesos = [12, 2, 1, 4]
14     valores = [4, 2, 2, 10]
15     capacidade_mochila = 15
16
17     # Inicialização do solver
18     solver = pywraplp.Solver.CreateSolver('SCIP')
19
20     # Variáveis
21     x = [solver.IntVar(0, 1, 'x[%i]' % i) for i in range(len(pesos))]
22
23     # Função objetivo
24     solver.Maximize(solver.Sum(x[i] * valores[i] for i in range(len(pesos))))
25
26     # Restrição de capacidade
27     solver.Add(solver.Sum(x[i] * pesos[i] for i in range(len(pesos))) <= capacidade_mochila)
28
29     while True:
30         # Resolvendo
31         status = solver.Solve()
32         if status == pywraplp.Solver.OPTIMAL:
33             print('Valor ótimo =', solver.Objective().Value())
34             break
35         else:
36             # Adicionando um plano de corte
37             expr = solver.Sum(x[i] for i in range(len(pesos)) if x[i].solution_value() > 0.5)
38             solver.Add(expr <= len([i for i in x if i.solution_value() > 0.5]) - 1)
39
40     mochila_planos_corte()
41
```

Figura 4.2 – Problema da mochila no Python

A Figura 4.2 apresenta um código para resolver iterativamente o problema da mochila, adicionando planos de corte quando a solução atual não é ótima, restringindo a busca por soluções melhores.

Para modelar problemas de programação linear, utiliza-se o Python, e suas diversas bibliotecas como PuLP, Gurobi, CPLEX ou SciPy. Elas oferecem suporte para a formulação e resolução de problemas de programação inteira e mista, permitindo a aplicação de métodos de enumeração e planos de corte para encontrar soluções ótimas ou aproximadas.

4.3.3 Branch and Bound

O método conhecido como Branch-and-Bound, de acordo com [2] é fundamentado na ideia de realizar uma enumeração inteligente dos pontos que podem ser soluções ótimas inteiras para um determinado problema. O termo *branch* refere-se à realização de partições no espaço das soluções. Por outro lado, o termo *bound* enfatiza a utilização de limites calculados durante a enumeração para comprovar a optimalidade da solução.

O funcionamento do método, que pode ser encontrado em [27] consiste inicialmente em considerar o espaço de soluções do problema. Ele começa com uma solução relaxada ou parcial e cria subproblemas dividindo o espaço de soluções em subconjuntos menores, chamados de "nós" ou "ramos". Em seguida, o algoritmo explora os nós em uma árvore de busca, geralmente começando pelo nó raiz, que representa o espaço de solução completo. Cada nó gera subnós correspondentes a novos subproblemas. Ao avaliar os subnós, o algoritmo pode utilizar técnicas de poda para cortar ramos que são improváveis de encontrar uma solução ideal. Isso é feito usando limites superiores (se uma solução parcial não pode ser melhorada) e limites inferiores (se uma solução parcial já é pior que a melhor solução conhecida). O processo continua, dividindo o espaço de soluções em subespaços menores e explorando cada nó da árvore de busca, descartando ramos não promissores. O algoritmo termina quando todos os ramos foram explorados ou quando é atingido um destino de parada, retornando a solução ótima ou a melhor solução encontrada. Um exemplo em que pode ser usado o método é o problema do caixeiro viajante, visto anteriormente na subseção 4.3.1.

4.3.4 Método de Gomory

O método de Gomory, introduzido por Ralph E. Gomory na década de 1950, destaca-se como uma técnica valiosa na resolução de problemas de programação linear inteira (ver [27]). Em tais problemas, as variáveis de decisão são limitadas a números inteiros, aumentando significativamente a complexidade em relação aos problemas de programação linear convencionais. Este método representa uma abordagem essencial para lidar com restrições lineares que envolvem variáveis inteiras, especialmente quando soluções fracionárias surgem durante a otimização. Seu objetivo principal é converter soluções fracionárias em soluções inteiras, refinando progressivamente o espaço de busca até alcançar a solução ótima. Ele está fundamentado no conceito de cortes de Gomory, os quais são planos de corte incorporados ao problema para eliminar soluções fracionárias indesejáveis. Esses cortes derivam-se dos conhecidos cortes de Gomory, que consistem em desigualdades lineares capazes de restringir as soluções a valores inteiros.

Em [27], são apresentados diversos exemplos ilustrativos do método de Gomory. Um exemplo típico seria a resolução de um problema de programação linear inteira onde as soluções fracionárias são um desafio. Durante a execução do método, soluções fracionárias são

verificadas, e se encontradas, são adicionados cortes de Gomory para reduzir o espaço de busca até que uma solução inteira ótima seja alcançada.

Além disso, o método de Gomory é amplamente aplicável em problemas de programação linear inteira, incluindo o problema de alocação de salas de aula. Em geral, quando um problema possui restrições lineares e variáveis de decisão que precisam ser inteiras, o método de Gomory pode ser utilizado para converter soluções fracionárias em soluções inteiras. No caso específico da alocação de salas de aula, onde as variáveis de decisão podem representar a alocação de aulas em salas com capacidades específicas, o método de Gomory poderia ser usado para garantir que as soluções encontradas sejam soluções inteiras viáveis, respeitando as restrições de capacidade das salas, os horários das aulas e outras restrições relevantes ao problema.

4.3.5 Branch and cut

O método Branch and Cut é uma extensão do método Branch and Bound, utilizado para resolver problemas de programação inteira mista. Ele combina a estratégia de divisão e poda (Branch and Bound) com a aplicação de cortes adicionais (cuts), que são desigualdades válidas que restringem o espaço de busca do problema.

O processo de funcionamento do método Branch and cut se inicia da mesma forma que o Branch and Bound, dividindo o espaço de soluções em subespaços menores. Além da estratégia de divisão, o método incorpora uma geração de cortes que são desigualdades válidas para restringir o espaço de busca. Essas seções são ampliadas ao problema para remover áreas do espaço de solução que não contêm uma solução ideal. O algoritmo continua a explorar os nós da árvore de busca, aplicando cortes e podas para eliminar ramos não promissores. Diferentes tipos de cortes podem ser usados, como cortes de Gomory, cortes de cobertura, cortes de clique, entre outros, dependendo do problema em questão. O método continua o processo até encontrar a solução ótima ou até um sorteio de parada a ser atingido.

Um problema comumente abordado em [25] usando o método Branch and Cut é o Problema de Roteamento de Veículos. Neste problema, uma frota de veículos precisa visitar um conjunto de clientes, minimizando a distância total percorrida por todos os veículos. O método em questão pode ser aplicado da seguinte maneira:

- **Branching:** No problema de roteamento de veículos, o método pode começar com uma formulação de programação inteira que define as rotas para os veículos. Ele pode dividir as rotas em subrotas menores ou em diferentes atribuições de clientes para diferentes veículos.
- **Corte (Cutting):** Os cortes podem ser adicionados para eliminar soluções inviáveis, como rotas que excedem a capacidade dos veículos ou rotas que não visitam todos os clientes.
- **Heurísticas:** Assim como no problema do caixeiro viajante, heurísticas podem ser usadas para acelerar a busca por soluções viáveis.

Branch and Cut é um método poderoso para resolver problemas de otimização combinatoria, pois utiliza a combinação de estratégias de ramificação, limitação e aplicação de cortes para reduzir o espaço de busca e encontrar soluções de alta qualidade de maneira eficiente.

4.3.6 Benders

O trabalho de métodos de decisão para lidar com problemas lineares inteiros mistos de ampla escala é uma prática comum. Esses métodos capitalizam a estrutura do modelo para permitir a fragmentação do problema em partes menores, abordando-as separadamente. Uma análise concisa das estruturas observações em questões práticas oferecem uma visão abrangente, aprimorando a compreensão e viabilizando soluções inovadoras para os desafios apresentados.

O Método de Decomposição de Benders é uma técnica clássica usada para resolver problemas lineares de grande porte. Inicialmente desenvolvida para enfrentar desafios complexos de programação linear inteira-mista, esse método tem sido bem executado em aplicações mais amplas. O processo envolve a complexidade das variáveis do problema em dois conjuntos, em que os valores das variáveis são estabelecidos no primeiro, simplificando assim o segundo conjunto para um problema linear mais simples, de fácil resolução. Essa metodologia tem demonstrado a capacidade de reduzir consideravelmente o tempo necessário para encontrar uma solução ótima.

O funcionamento do Método de Benders é apresentado a seguir, baseando-se na explicação detalhada encontrada em [27]

- Inicialmente, o problema é dividido em duas partes: um problema principal (mestre) e um conjunto de subproblemas (chamados de problemas de corte ou subproblemas de Benders).
- O problema principal é um problema menor que contém parte das variáveis do problema original e é resolvido para encontrar uma solução inicial ou parcial.
- Cada subproblema é um problema menor que aborda um aspecto específico do problema original. Esses subproblemas são resolvidos iterativamente para gerar informações chamadas de "cortes" ou "planos de corte" que são restrições adicionais válidas para o problema principal.
- Os cortes gerados pelos subproblemas são acrescentados ao problema principal, restringindo o espaço de busca.
- O processo continua iterativamente, com a resolução do problema principal e a geração de cortes pelos subproblemas. A cada iteração, as cortes adicionais ajudaram a melhorar a solução do problema principal.
- O método continua até que a solução do problema principal atinja uma convergência satisfatória ou um sorteio de parada seja realizado.

Um exemplo que é aplicado o método de benders é o problema de fluxos de redes, visto em [2]. Os problemas de fluxo são centralizados na otimização da distribuição de produtos que se originam em pontos de oferta e são consumidos em pontos de demanda, dentro de redes com interconexões possíveis. Esses problemas são comumente encontrados em ambientes industriais, sistemas de comunicação, transporte e distribuição de água, mas também funcionam como modelos para uma ampla gama de outras situações semelhantes por meio de abstração. Cada produto tem sua oferta e demanda conhecidas, não sendo obrigatório que o fluxo ocorra diretamente de um ponto de produção para um ponto de demanda, permitindo a inclusão de pontos intermediários, como depósitos ou centros de distribuição. As interconexões podem estar sujeitas a restrições de capacidade de tráfego e diferentes custos associados.

Conforme descrito por [2], uma formulação matemática restrita se aplica ao problema de fluxo de custo mínimo em uma rede composta por $n = m + 2$ nós, onde m representa o número de nós reais e n é o número de nós na rede equilibrada. Este problema pode ser formulado da seguinte maneira:

$$\text{Minimizar } z = \sum_{(i,j) \in E} c_{ij} x_{ij}$$

sujeito a:

$$\sum_{(i,j) \in E} x_{ij} - \sum_{(k,i) \in E} x_{ki} = 0 \quad i = 1, \dots, n$$

$$l_{ij} \leq x_{ij} \leq L_{ij} \quad \forall (i,j) \in E$$

Os problemas em redes podem ser classificados, de um modo geral, dentro da seguinte hierarquia, extraída de [2].

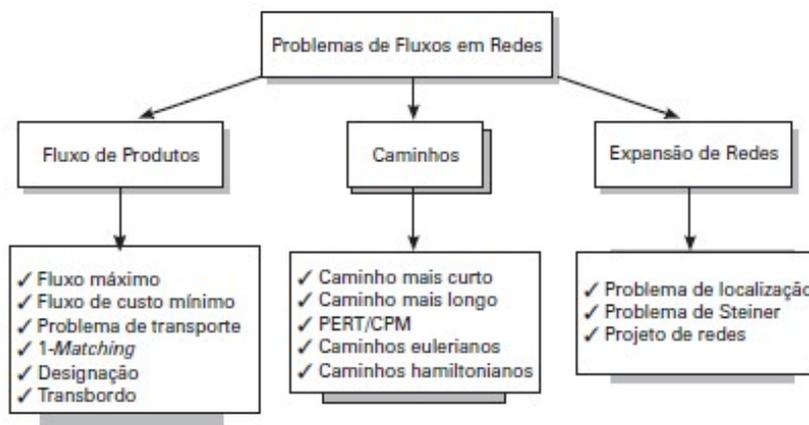


Figura 4.3 – Taxonomia para problemas de fluxo. Extraído de [2], em 09/12/2023

Modelos de gestão operacional de restaurantes e de salas de aula em centros universitários e simulações computacionais

5.1 Modelo de gestão operacional de restaurantes universitários

Nesta seção é destacado um modelo de otimização linear inteira mista, baseado no modelo da dieta, cujo objetivo é representar a realidade do Restaurante Universitário do *campus* Nova Iguaçu da Universidade Federal Rural do Rio de Janeiro. Tal modelo representa um melhoramento do modelo apresentado em [6], tendo em vista que no presente trabalho são considerados alimentos cujo as porções são servidas em quantidades inteiras.

Primeiramente, uma refeição, como o café da manhã, almoço, jantar ou qualquer outra refeição fornecida a um público considerável que requer um mínimo de planejamento ao longo do dia em um restaurante universitário (RU) é fixada. Usaremos quatro índices para identificar as porções a serem fornecidas, a saber, \bar{i} , \bar{j}_i , \tilde{j}_i e k , onde \bar{i} indica o grupo ao qual a comida pertence, \bar{j}_i representa o alimento cujo a quantidade de porções pode ser expressa numericamente por valores fracionários não negativos, enfatizando seu grupo, \tilde{j}_i representa o alimento cujo a quantidade de porções pode ser expressa numericamente apenas por valores inteiros não negativos, também enfatizando seu grupo, e k denota o dia da semana no qual a comida será fornecida.

Considere a existência de \bar{N} grupos de alimentos catalogados, em que o \bar{i} -ésimo grupo contém exatamente \bar{M}_i alimentos cujo a quantidade de porções assume valores fracionários não negativos e exatamente \tilde{M}_i alimentos cujo a quantidade de porções assume apenas valores inteiros não negativos. Além disso, considere que o restaurante opera ao longo de P dias da semana. Denotamos por $y_{\bar{j}_i k}$ e $y_{\tilde{j}_i k}$ as quantidades de porções dos alimentos \bar{j}_i e \tilde{j}_i , respectivamente, do grupo \bar{i} , a serem fornecidas no k -ésimo dia, na refeição fixada. Nosso objetivo inicial consiste em minimizar o custo associado ao fornecimento semanal da refeição fixada para os estudantes.

Como as porções unitárias dos alimentos \bar{j}_i e \tilde{j}_i do grupo \bar{i} tem custo $\bar{c}_{\bar{j}_i}$ e $\bar{c}_{\tilde{j}_i}$, respectivamente, relacionados a si, os custos associados ao fornecimento de $y_{\bar{j}_i k}$ e $y_{\tilde{j}_i k}$ porções deles, no k -ésimo dia, são $\bar{c}_{\bar{j}_i} y_{\bar{j}_i k}$ e $\bar{c}_{\tilde{j}_i} y_{\tilde{j}_i k}$. Considerando que o modelo é proposto para representar a operação semanal de um RU ao longo de P dias, a função objetivo é resumida como:

$$\sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_i=1}^{\bar{M}_i} \sum_{k=1}^P \bar{c}_{\bar{j}_i} y_{\bar{j}_i k} + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\tilde{j}_i=1}^{\tilde{M}_i} \sum_{k=1}^P \bar{c}_{\tilde{j}_i} y_{\tilde{j}_i k}. \quad (5.1)$$

Em outras palavras, temos um par de somatórios duplos para cada grupo envolvendo os produtos

$\bar{c}_{\bar{j}_i} y_{\bar{j}_i k}$ e $\tilde{c}_{\bar{j}_i} y_{\bar{j}_i k}$, respectivamente, totalizando um total de \bar{N} pares de somatórios duplos, onde $\bar{j}_i = 1, \dots, \bar{M}_i$, $\tilde{j}_i = 1, \dots, \tilde{M}_i$ e $k = 1, \dots, P$.

O primeiro conjunto de restrições diz respeito à quantidade de porções de alimentos do mesmo grupo fornecidas na refeição fixada, em um determinado dia. Como cada grupo de alimentos é representado por um índice \bar{i} ($\bar{i} = 1, \dots, \bar{N}$), podemos estabelecer um limite superior para a quantidade de porções de alimentos do mesmo grupo. Denotamos isso por $\bar{Q}_{max}^{\bar{i}}$. Além disso, podemos exigir que pelo menos $\bar{Q}_{min}^{\bar{i}}$ porções de alimentos de cada grupo sejam fornecidas na refeição definida em um determinado dia. Assim, no k -ésimo dia, para algum $k = 1, \dots, P$, temos:

$$\bar{Q}_{min}^{\bar{i}} \leq \sum_{\bar{j}_i=1}^{\bar{M}_i} y_{\bar{j}_i k} + \sum_{\tilde{j}_i=1}^{\tilde{M}_i} y_{\tilde{j}_i k} \leq \bar{Q}_{max}^{\bar{i}}; \quad \bar{i} = 1, \dots, \bar{N}, \quad k = 1, \dots, P. \quad (5.2)$$

Note que (5.2) representa um total de $\bar{N} \cdot P$ restrições a serem incorporadas no modelo.

Para o segundo conjunto de restrições, investigamos a correspondência entre alimentos e quantidades de cada nutriente presente em suas respectivas composições, bem como os limites diários de ingestão desses nutrientes na refeição fixada. Suponha que seja conhecida uma lista de \bar{R} nutrientes para uma dieta saudável. Denotamos por $a_{\bar{j}_i r}$ e $a_{\tilde{j}_i r}$ as frações do r -ésimo nutriente presente em cada porção dos alimentos \bar{j}_i e \tilde{j}_i , respectivamente, do grupo \bar{i} , e por \bar{A}_{min}^r e \bar{A}_{max}^r os limites inferiores e superiores de ingestão para o nutriente r , respectivamente, na refeição definida. Segue que

$$\bar{A}_{min}^r \leq \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_i=1}^{\bar{M}_i} a_{\bar{j}_i} y_{\bar{j}_i} k + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\tilde{j}_i=1}^{\tilde{M}_i} a_{\tilde{j}_i} y_{\tilde{j}_i} k \leq \bar{A}_{max}^r; \quad k = 1, \dots, P, \quad r = 1, \dots, \bar{R}. \quad (5.3)$$

O conjunto de restrições (5.3) contém exatamente $P \cdot \bar{R}$ desigualdades duplas. Vale ressaltar que os limites \bar{A}_{min}^r e \bar{A}_{max}^r correspondem às frações dos limites diários de ingestão de nutrientes. Geralmente, eles são recomendados previamente por nutricionistas para cada refeição.

O terceiro conjunto de restrições refere-se ao valor calórico da refeição fixada. O cardápio da refeição fixada está baseado nas quilocalorias (kcal) fornecidas pelos alimentos, estabelecendo um consumo de pelo menos \bar{B}_{min} kcal e no máximo \bar{B}_{max} kcal. Denotamos por $b_{\bar{j}_i}$ e $b_{\tilde{j}_i}$ os valores calóricos associados às porções dos alimentos \bar{j}_i -ésimo e \tilde{j}_i -ésimo, respectivamente, do grupo \bar{i} , a serem consumidos na refeição fixada. Assim, resumimos o seguinte conjunto de restrições:

$$\bar{B}_{min} \leq \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_i=1}^{\bar{M}_i} b_{\bar{j}_i} y_{\bar{j}_i} k + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\tilde{j}_i=1}^{\tilde{M}_i} b_{\tilde{j}_i} y_{\tilde{j}_i} k \leq \bar{B}_{max}; \quad k = 1, \dots, P. \quad (5.4)$$

Se desejamos garantir que pelo menos $\bar{Q}_{min}^{\bar{j}_i}$ e $\bar{Q}_{min}^{\tilde{j}_i}$ e no máximo $\bar{Q}_{max}^{\bar{j}_i}$ e $\bar{Q}_{max}^{\tilde{j}_i}$ porções dos alimentos \bar{j}_i -ésimo e \tilde{j}_i -ésimo, respectivamente, do grupo \bar{i} , sejam fornecidas ao longo da semana, então precisamos incluir apenas o seguinte conjunto de desigualdades duplas:

$$\bar{Q}_{min}^{\bar{j}_i} \leq \sum_{k=1}^P y_{\bar{j}_i} k \leq \bar{Q}_{max}^{\bar{j}_i}, \quad (5.5)$$

$$\bar{Q}_{min}^{\tilde{j}_i} \leq \sum_{k=1}^P y_{\tilde{j}_i} k \leq \bar{Q}_{max}^{\tilde{j}_i}. \quad (5.6)$$

Perceba que (5.5) e (5.6), respectivamente, compreendem não mais do que $\sum_{\bar{i}=1}^{\bar{N}} \bar{M}_{\bar{i}}$ e

$\sum_{\bar{i}=1}^{\bar{N}} \tilde{M}_{\bar{i}}$ restrições cada uma, já que nem todos os alimentos cumprem com elas. Além disso, se

exatamente $\bar{Q}^{\bar{j}_i}$ e $\tilde{Q}^{\bar{j}_i}$ porções dos alimentos específicos \bar{j}_i e \tilde{j}_i devem ser fornecidas na semana, então, ao definirmos $\bar{Q}^{\bar{j}_i} = \bar{Q}_{min}^{\bar{j}_i} = \bar{Q}_{max}^{\bar{j}_i}$ e $\tilde{Q}^{\bar{j}_i} = \tilde{Q}_{min}^{\bar{j}_i} = \tilde{Q}_{max}^{\bar{j}_i}$, elas serão garantidas.

Por fim, introduzimos restrições não negativas (ou restrições canônicas) para as variáveis de decisão:

$$y_{\bar{j}_i k} \geq 0 (y_{\bar{j}_i k} \in \mathbb{R}), y_{\tilde{j}_i k} \geq 0 (y_{\tilde{j}_i k} \in \mathbb{Z}); \bar{i} = 1, \dots, \bar{N}, \bar{j}_i = 1, \dots, \bar{M}_i, \tilde{j}_i = 1, \dots, \tilde{M}_i, k = 1, \dots, P. \quad (5.7)$$

O modelo de gerenciamento de custos envolvidos na refeição fixada consiste em minimizar a função objetivo (5.1) sujeita a o agrupamento de restrições listadas de (5.2) a (5.7).

Agora, suponha que o \bar{i} -ésimo grupo tenha seus alimentos reorganizados em subgrupos, não necessariamente disjuntos, identificados por algum número inteiro \bar{l} ($\bar{l} = 1, \dots, T$), que representa a refeição na qual será fornecido. Sejam $\bar{M}_{\bar{i}\bar{l}}$ e $\tilde{M}_{\bar{i}\bar{l}}$ ($\bar{i} = 1, \dots, \bar{N}$, $\bar{l} = 1, \dots, T$) as quantidades de alimentos a serem fornecidas em porções de tamanhos fracionário e inteiro, respectivamente, no \bar{i} -ésimo grupo, disponível para ser fornecido na \bar{l} -ésima refeição. Denotemos ainda por $y_{\bar{j}_{i\bar{l}}k}$ e $y_{\tilde{j}_{i\bar{l}}k}$ as variáveis que indicam quantidades porções de tamanhos fracionário e inteiro dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$, respectivamente, do grupo \bar{i} , disponíveis para serem fornecidas na \bar{l} -ésima refeição, em um determinado dia. Além disso, denotemos por $(\bar{c}_{\bar{j}_{i\bar{l}}}, \bar{c}_{\tilde{j}_{i\bar{l}}})$, $(a_{\bar{j}_{i\bar{l}}r}, a_{\tilde{j}_{i\bar{l}}r})$, $(b_{\bar{j}_{i\bar{l}}}, b_{\tilde{j}_{i\bar{l}}})$ os pares de custos unitários associados a porções de tamanhos fracionário e inteiro dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$, respectivamente, de frações do r -ésimo nutriente ($r = 1, \dots, \bar{R}$) presentes em porções de tamanhos fracionário e inteiro dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$, respectivamente e de quantidades, em kcal, presentes em porções de tamanhos fracionário e inteiro dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$, respectivamente. Denote ainda por $\bar{A}_{min}^{r\bar{l}}$ e $\bar{A}_{max}^{r\bar{l}}$ os limites inferiores e superiores de ingestão do l -ésimo nutriente na \bar{l} -ésima refeição dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$. adicionalmente, denote por $\bar{B}_{min}^{\bar{l}}$ e $\bar{B}_{max}^{\bar{l}}$ os limites de consumo energético, em kcal, necessários para um estudante na \bar{l} -ésima refeição. Finalmente, denote por $(\bar{Q}_{min}^{\bar{j}_{i\bar{l}}}, \tilde{Q}_{min}^{\bar{j}_{i\bar{l}}})$ e $(\bar{Q}_{max}^{\bar{j}_{i\bar{l}}}, \tilde{Q}_{max}^{\bar{j}_{i\bar{l}}})$ os pares de limites inferior e superior para quantidades de porções de tamanhos fracionário e inteiro dos alimentos $\bar{j}_{i\bar{l}}$ e $\tilde{j}_{i\bar{l}}$, respectivamente, a serem fornecidos em uma semana. Uma vez que o objetivo atual consiste em minimizar o custo total envolvido no fornecimento semanal de refeições para um estudante com um perfil estabelecido por um(a) nutricionista, o problema linear inteiro misto, denotado por (WSMS), resume-se a minimizar (5.8), que representa uma extensão de (5.1), sujeito às restrições (5.9) a (5.14), que representam generalizações das restrições (5.2) a (5.7), conforme resumido a seguir:

$$\min \quad \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{l}=1}^T \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} \sum_{k=1}^P \bar{c}_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{l}=1}^T \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} \sum_{k=1}^P \bar{c}_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k, \quad (5.8)$$

$$\text{s.t.} \quad Q_{min}^{\bar{i}} \leq \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k + \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k \leq Q_{max}^{\bar{i}}; \bar{i} = 1, \dots, \bar{N}, \bar{l} = 1, \dots, T, k = 1, \dots, P. \quad (5.9)$$

$$\bar{A}_{min}^{\bar{r}\bar{l}} \leq \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} a_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} a_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k \leq \bar{A}_{max}^{\bar{r}\bar{l}}; \bar{l} = 1, \dots, T, k = 1, \dots, P, r = 1, \dots, \bar{R}. \quad (5.10)$$

$$\bar{B}_{min}^{\bar{l}} \leq \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} b_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k + \sum_{\bar{i}=1}^{\bar{N}} \sum_{\bar{j}_{\bar{i}\bar{l}}=1}^{\bar{M}_{\bar{i}\bar{l}}} b_{\bar{j}_{\bar{i}\bar{l}}} y_{\bar{j}_{\bar{i}\bar{l}}}^k \leq \bar{B}_{max}^{\bar{l}}; \bar{l} = 1, \dots, T, k = 1, \dots, P. \quad (5.11)$$

$$\bar{Q}_{min}^{\bar{j}_{\bar{i}\bar{l}}} \leq \sum_{k=1}^P y_{\bar{j}_{\bar{i}\bar{l}}}^k \leq \bar{Q}_{max}^{\bar{j}_{\bar{i}\bar{l}}}. \quad (5.12)$$

$$\bar{Q}_{min}^{\bar{j}_{\bar{i}\bar{l}}} \leq \sum_{k=1}^P y_{\bar{j}_{\bar{i}\bar{l}}}^k \leq \bar{Q}_{max}^{\bar{j}_{\bar{i}\bar{l}}}. \quad (5.13)$$

$$y_{\bar{j}_{\bar{i}\bar{l}}}^k, y_{\bar{j}_{\bar{i}\bar{l}}}^k \geq 0; y_{\bar{j}_{\bar{i}\bar{l}}}^k \in \mathbb{R}, y_{\bar{j}_{\bar{i}\bar{l}}}^k \in \mathbb{Z}. \quad (5.14)$$

5.2 Modelo de gestão de salas de aula em centros universitários

O modelo de alocação de salas de aula apresentado aqui representa um melhoramento do modelo apresentado em [7] pois realiza um ajuste em um dos conjuntos de restrições lá destacados, a saber, o conjunto de restrições (5.19). Além disso, o presente modelo também engloba as restrições que permitem que cada departamento aloque até 20% de disciplinas, do total de disciplinas ofertadas, em um turno que não coincide com aquele no qual ele geralmente oferta suas disciplinas, isto é, que coincide com o turno de funcionamento do curso cujo a maior parte dos docentes integram o seu colegiado. No entanto, estas restrições, em específico, só serão discutidas e apresentadas ao final da Seção 5.4. Vale ressaltar também que no presente modelo o objetivo consiste em maximizar a cobertura discente, isto é, maximizar a quantidade de alunos inscritos em disciplinas que de fato estão alocadas em alguma sala.

Vamos introduzir as variáveis de decisão, bem como a função objetivo e as restrições associadas à aplicação do modelo de alocação de salas de aula à alocação semanal de disciplinas em salas de aula em centros universitários. Para este caso, algumas premissas foram previamente assumidas, entre as quais se destacam as seguintes: disponibilidade conhecida de salas de aula por semestre acadêmico; prioridade de alocação de disciplinas com o maior número de estudantes matriculados; número fixo de dias úteis para a alocação; alocação de disciplinas teórica de, no máximo, W créditos acadêmicos, para algum número inteiro positivo W ; salas de aula que suportam apenas uma disciplina em um dia específico, turno e horário.

Geralmente, disciplinas práticas são atribuídas a laboratórios cujo a gestão é de responsabilidade de departamentos universitários específicos. Assim, tanto laboratórios quanto disciplinas práticas não são considerados em nossa formulação. Além disso, uma mesma disciplina que é oferecida a duas turmas distintas, seja no mesmo horário ou em horários diferentes, é compreendida pelo modelo como, na verdade, duas disciplinas e isso é explicitado no total de disciplinas oferecidas.

Suponha que existam N disciplinas a serem alocadas em M salas de aula ao longo de P dias úteis, destinados a aulas. Além disso, suponha que cada dia seja dividido em Q turnos

e cada turno contenha R horários. Utilizamos cinco índices para a definição das variáveis de decisão, a saber: o índice i ($i = 1, \dots, N$) para rotular disciplinas, j ($j = 1, \dots, M$) para rotular salas de aula, k ($k = 1, \dots, P$) para os dias úteis, l ($l = 1, \dots, Q$) para os turnos e m ($m = 1, \dots, R$) para os horários.

Defina por x_{ijklm} a variável de decisão associada à alocação (ou não) da i -ésima disciplina na j -ésima sala de aula, no k -ésimo dia útil, no m -ésimo horário do l -ésimo turno. Assim,

$$x_{ijklm} = \begin{cases} 1, & \text{se a alocação ocorre,} \\ 0, & \text{caso contrário.} \end{cases}$$

Seja c_i o total de estudantes matriculados na i -ésima disciplina. Se $x_{ijklm} = 1$, então exatamente c_i estudantes estão sendo cobertos pela j -ésima sala de aula, no k -ésimo dia, no m -ésimo horário do l -ésimo turno. Note que se a mesma disciplina, i , for alocada em m horários de um mesmo turno l , então o número de estudantes cobertos, neste turno, deve permanecer igual a c_i . Por outro lado, o número de horários alocados para a i -ésima disciplina é dado por $\sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R x_{ijklm}$. Uma vez que nosso objetivo consiste em maximizar o número de estudantes matriculados em alguma disciplina alocada em uma sala de aula, ao fazer

$$z_{ijklm} = \begin{cases} \frac{x_{ijklm}}{\sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R x_{ijklm}} \cdot c_i, & \text{se } \sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R x_{ijklm} \geq 1 \\ 0, & \text{caso contrário,} \end{cases} \quad (5.15)$$

o valor objetivo Z pode ser resumido como:

$$Z = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R z_{ijklm}. \quad (5.16)$$

A linearidade do problema de atribuição de salas de aula é perdida quando a expressão (5.16) é incorporada ao modelo como função objetivo. Ao contrário disso, assumimos a seguinte simplificação para o valor objetivo Z

$$Z = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R c_i x_{ijklm}, \quad (5.17)$$

e, após uma solução ótima x_{ijklm}^* ser computada, o valor ótimo Z^* é ajustado para $Z^* = \sum_{i=1}^N \sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R z_{ijklm}^*$, onde z_{ijklm}^* é o valor de z_{ijklm} , calculado em x_{ijklm}^* .

Para garantir que, no máximo, a i -ésima disciplina será atribuída à j -ésima sala de aula, no k -ésimo dia útil, no m -ésimo horário do l -ésimo turno, uma vez que nenhuma outra disciplina pode ser atribuída à j -ésima sala de aula, respeitando os mesmos requisitos, devemos assumir o seguinte conjunto de restrições:

$$\sum_{i=1}^N x_{ijklm} \leq 1; \quad j = 1, \dots, M, \quad k = 1, \dots, P, \quad l = 1, \dots, Q, \quad m = 1, \dots, R. \quad (5.18)$$

Além disso, cada turno tem R horários. Para que uma disciplina tenha dois horários consecutivos atribuídos no mesmo turno, devemos introduzir o seguinte conjunto de restrições:

$$\begin{aligned} x_{ijklm} - x_{ijkl(m+1)} &= 0, \text{ se } m \text{ é ímpar;} \\ x_{ijklm} + x_{ijkl(m+1)} &\leq 1, \text{ caso contrário;} \\ i &= 1, \dots, N, j = 1, \dots, M, k = 1, \dots, P, l = 1, \dots, Q, m = 1, \dots, R-1. \end{aligned} \tag{5.19}$$

Observe que dois horários consecutivos são alocados ou não quando as variáveis correspondentes assumem o mesmo valor (0, se não alocados, ou 1, caso contrário). Por outro lado, o conjunto de restrições (5.19) implica necessariamente que a alocação de apenas um crédito acadêmico de uma disciplina em um turno fixado, de um dia também fixado, para disciplinas com um número ímpar de créditos acadêmicos, ocorre apenas em horários pares. Isso não é uma desvantagem do modelo se a quantidade de disciplinas com um número ímpar de créditos acadêmicos for pequena. Além disso, este único crédito acadêmico poderia ser atribuído a um laboratório, como é feito com disciplinas de um crédito acadêmico, ou poderia ser facilmente resolvido por meio da implementação de ensino híbrido, sendo alocado em uma sala virtual para operar no formato remoto, caso a alocação de apenas um crédito acadêmico em horários pares se torne um impicílio.

Seja w ($w = 1, \dots, W$) um índice inteiro não negativo para denotar a quantidade de créditos acadêmicos associados às disciplinas. Além disso, defina por $I_w \subset \{1, \dots, N\}$ o conjunto de índices de disciplinas com w créditos acadêmicos. Para as restrições que seguem, devemos admitir que:

$$\sum_{j=1}^M \sum_{k=1}^P \sum_{l=1}^Q \sum_{m=1}^R x_{ijklm} \leq w; \quad i \in I_w, \quad w = 1, \dots, W, \tag{5.20}$$

Perceba que o conjunto de restrições (5.20) não garante que todos os créditos acadêmicos associados a uma disciplina sejam realmente alocados em uma sala de aula, já que a quantidade de salas de aula pode ser insuficiente para atender à demanda apresentada. No entanto, (5.20) permite identificar a quantidade exata de créditos acadêmicos associados a uma disciplina que não foi alocada em uma sala de aula, além de ajudar a prever a quantidade mínima de salas de aula excedentes necessárias para atender integralmente à demanda apenas adicionando novas salas de aula e refazendo a simulação.

Além disso, devemos identificar tanto a quantidade de créditos acadêmicos associados a uma disciplina quanto o turno no qual ela deve ser alocada, caso deva ser especificamente alocada em um turno específico. Denote por w_i o número de créditos acadêmicos associados à i -ésima disciplina e por $I_l \subset \{1, \dots, N\}$ o conjunto de índices de disciplinas que devem ser obrigatoriamente atribuídas ao l -ésimo turno. Então:

$$\sum_{j=1}^M \sum_{k=1}^P \sum_{m=1}^R x_{ijklm} \leq w_i; \quad i \in I_l, \quad l = 1, \dots, Q. \tag{5.21}$$

Os créditos acadêmicos associados a uma disciplina que não foram alocados em uma sala de aula, em um turno obrigatório, podem ser alocados em uma sala de aula em outro turno. Isso justifica o uso de desigualdades e não igualdades em (5.21), além de evitar a necessidade de salas de aula excedentes para alocação. De fato, o conjunto de restrições (5.21) é mais específico do que o conjunto de restrições (5.20).

Além disso, para disciplinas com um número de créditos acadêmicos que excede 2, considerando o tempo de concentração dos estudantes que, em geral, não é maior do que o

tempo associado a dois horários consecutivos, em um mesmo turno de um mesmo dia, podemos definir, para um determinado turno l , de um dia k , que:

$$\sum_{j=1}^M \sum_{m=1}^R x_{ijklm} \leq 2; \quad i \in I_l, \quad k = 1, \dots, P, \quad l = 1, \dots, Q, \quad w_i \geq 2. \quad (5.22)$$

Agora, admita que as M salas de aula disponíveis podem ser classificadas em S grupos, onde o s -ésimo grupo ($s = 1, \dots, S$) é composto por salas de aula que acomodam no máximo C_s estudantes. Defina por $J_s \subset \{1, \dots, M\}$ o conjunto de índices das salas de aula que acomodam no máximo C_s estudantes. Segue que:

$$\sum_{i=1}^N c_i x_{ijklm} \leq C_s; \quad j \in J_s, \quad k = 1, \dots, P, \quad l = 1, \dots, Q, \quad m = 1, \dots, R, \quad s = 1, \dots, S. \quad (5.23)$$

Observe que o conjunto de restrições (5.23) abrange aspectos inerentes ao modelo da mochila. Finalmente, adicionamos restrições de valor binário para as variáveis de decisão:

$$x_{ijklm} \in \{0, 1\}; \quad i = 1, \dots, N, \quad j = 1, \dots, M, \quad k = 1, \dots, P, \quad l = 1, \dots, Q, \quad m = 1, \dots, R. \quad (5.24)$$

Portanto, o modelo inteiro binário, representado aqui por (WSAC), consiste em maximizar (5.17), sujeito às restrições (5.18) a (5.24).

5.3 Descrição de cenário para o modelo de gestão de restaurantes universitários aplicado ao RU-IM/UFRRJ

Para a modelagem, os dados foram coletados no Restaurante Universitário do Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro (RU-IM/UFRRJ), Campus de Nova Iguaçu, no segundo semestre acadêmico de 2019, precisamente o último semestre de operação antes da Pandemia do Coronavírus.

Em média, o RU-IM/UFRRJ oferecia aproximadamente 1.050 refeições aos estudantes por turno, nomeadamente manhã, tarde e noite. Sua operação semanal começava na segunda-feira e terminava na sexta-feira. Com relação aos dados coletados para o segundo semestre acadêmico de 2019, os cardápios diários eram distribuídos para café da manhã ($\bar{l} = 1$), almoço ($\bar{l} = 2$) e jantar ($\bar{l} = 3$), com uma média diária de presença de 2.053 alunos. É importante enfatizar que o RU-IM não tem como objetivo o lucro com as refeições oferecidas aos alunos, mas sim a possibilidade de oferecer uma dieta completa e saudável, fornecendo refeições que custam um valor monetário negligenciável em comparação com valores de outras refeições com as mesmas porções de comida e níveis nutricionais oferecidos por restaurantes privados.

Com relação ao público-alvo, as refeições são montadas tendo como referência o Programa de Alimentação do Trabalhador (PAT) [28], que foi desenvolvido no Brasil por meio da Lei nº 6.321, de 14 de abril de 1976, e regulamentado pelo Decreto nº 5 de 14 de abril de 1991. O PAT sugere que as refeições para trabalhadores com renda mensal de até cinco salários mínimos recebam alimentação por meio de sua relação de emprego (refeição pronta ou cesta básica), e que sejam compostas por 55 a 60% de carboidratos, 10 a 15% de proteínas e 25 a 30% de lipídios. Como os níveis nutricionais são projetados por refeição oferecida no RU-IM/UFRRJ, é fácil perceber que haverá um padrão PAT para café da manhã, um para o almoço e um para o jantar, respectivamente, ou seja, cada refeição oferecida em seu respectivo turno terá seus níveis de nutrientes planejados individualmente.

O perfil padrão do aluno é descrito como um adulto com uma massa corporal de 70 kg, aproximadamente, e que consome ao longo do dia 1500kcal, distribuídas em 300kcal para o café da manhã e 600 a 700kcal para o almoço ou jantar, em média, respectivamente. Nutrientes como carboidratos (simples e complexos), proteínas (animal e vegetal), lipídios, vitaminas e minerais (ferro, vitamina A, vitamina D e sódio) são garantidos ao longo do dia.

As refeições fornecidas apresentam cardápios. O café da manhã tem seus itens alimentares classificados e separados por quatro grupos, a saber, *bebidas* ($\bar{i} = 1$), *pães, bolos ou biscoitos* ($\bar{i} = 2$), *frutas* ($\bar{i} = 3$) e *complementos* ($\bar{i} = 4$). Alimentos para o almoço ou jantar são agrupados em *proteínas animais* ($\bar{i} = 1$), *proteínas alternativas* ($\bar{i} = 2$), *proteínas vegetais* ($\bar{i} = 3$), *acompanhamentos* ($\bar{i} = 4$), *guarnições* ($\bar{i} = 5$), *saladas* ($\bar{i} = 6$), *sobremesas* ($\bar{i} = 7$) e *bebidas* ($\bar{i} = 8$).

Devido a limitação de recursos financeiros, é inviável para o RU-IM/UFRRJ repetir certos grupos de alimentos na mesma refeição para um aluno. Por exemplo, no almoço ou jantar, os alunos recebem apenas uma guarnição, seja uma proteína animal ou vegetal e uma porção de sobremesa. Além disso, os alimentos também são classificados em grupos para serem fornecidos em porções inteiras () ou fracionadas (), de acordo com a demanda.

Observe que durante um dia de operação no RU-IM/UFRRJ, apenas dois planos de cardápio são montados, um para o café da manhã e outro que cobre o almoço e o jantar (o mesmo cardápio oferecido no almoço será oferecido no jantar). Como resultado, os alimentos usados para produzir um dia de operação no RU são separados de acordo com cada necessidade para a refeição que será fornecida, garantindo assim um cardápio coerente para todo o público que o consome.

As tabelas 5.1 e 5.2 destacam valores por porção de cada item oferecido no café da manhã e almoço/jantar, respectivamente, do restaurante universitário.

Tabela 5.1 – Café da manhã

Quantidade	Item	Preço
(300mL)	Leite	1,50
(300mL)	Café	1,50
(300mL)	Achocolatado	1,50
(300mL)	Refresco	1,50
(1 un)	Pães	5,00
(1 un)	Bolo de milho	5,00
(1 un)	Biscoito	5,00
(1 un)	Maçã	1,00
(1 un)	Tangerina	1,15
(1 un)	Banana	1,25
(1 un)	Pera	1,54
(1 un)	Laranja	0,85
(10g)	Queijos	0,60
(10g)	Presunto	0,74
(25g)	Ovo Mexido	1,11
(10g)	Geleia	0,60
(10g)	Margarina	1,00
(10g)	Manteiga	1,12

Tabela 5.2 – Almoço/Jantar

Quantidade	Item	Preço
(100g)	Frango Assado	2,00
(100g)	Frango Grelhado	1,19
(100g)	Almondegas	2,12
(100g)	Carne Moída	2,22
(100g)	Hambúrguer Vegetariano	2,11
(100g)	Lentilha	2,15
(60g)	Arroz	1,71
(60g)	Arroz com brócolis	1,82
(60g)	Macarrão	1,67
(60g)	Feijão	1,70
(60g)	Farofa	1,52
(20g)	Alface	0,82
(20g)	Alface com tomate	0,93
(20g)	rúcula	0,85
(20g)	Beterraba	0,80
(20g)	Salada de Grão de Bico	1,11
(1 un)	Maçã (sobremesa)	1,00
(1 un)	Banana (sobremesa)	1,25
(1 un)	Laranja (sobremesa)	0,85
(300mL)	Guaraná	1,50
(300mL)	Uva	1,75
(300mL)	Groselha	1,51

5.4 Descrição de cenário para o modelo de gestão de salas de aula em centros universitários aplicado ao IM/UFRRJ

O Instituto Multidisciplinar da Universidade Federal Rural do Rio de Janeiro (IM/UFRRJ), no campus de Nova Iguaçu, oferece cursos de graduação em Administração, Ciência da Computação, Direito, Economia, Geografia, História, Letras, Matemática, Pedagogia e Turismo, ministrados por 9 departamentos: o Departamento de Administração e Turismo (DAT), o Departamento de Ciência da Computação (DCC), o Departamento de Ciências Econômicas (DPCE), o Departamento de Ciências Jurídicas (DCJUR), o Departamento de Educação e Sociedade (DES), o Departamento de Geografia (DGEO), o Departamento de História (DH), o Departamento de Letras (DL) e o Departamento de Tecnologias e Linguagens (DTL), além de disciplinas de cursos de pós-graduação.

As aulas acontecem em quatro blocos: Administrativo (ADM), Multidisciplinar (MUL), Informática (INF) e Biblioteca (BIB), e ocorrem em três turnos: manhã, tarde e noite, ao longo de cinco dias, de segunda a sexta-feira, de acordo com os requisitos apresentados por cada departamento. Além disso, o instituto dispõe de 31 salas de aula: 6 com capacidade máxima para 25 alunos (C_1), 20 com capacidade máxima para 50 alunos (C_2) e 5 com capacidade máxima para 75 alunos (C_3). Cada turno possui 4 horários. As aulas começam às 8h e terminam às 22h, com duração de uma hora cada. A Tabela 5.3 apresenta as quantidades de disciplinas oferecidas pelos departamentos, considerando os turnos (I) em que são oferecidas, a quantidade de créditos acadêmicos (w) associados a elas (ou seja, 2, 3 e 4) e o tamanho necessário das salas de aula (C_s , $s = 1, 2, 3$). Os dados disponíveis para a simulação consideraram o segundo semestre do

ano letivo de 2019, totalizando 402 disciplinas de graduação a serem distribuídas para atender a 96.721 alunos.

Tabela 5.3 – Dados de entrada para o modelo WSAC.

$\frac{l}{w}$ $\frac{C_s}{D}$	1									2									3									Total	
	2			3			4			2			3			4			2			3			4				
	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃	C ₁	C ₂	C ₃		C ₁
DAT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	0	0	1	0	3	59	0	73	
DCC	0	0	0	0	1	0	1	1	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	
DPCE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	29	0	32	
DCJUR	0	2	9	0	0	0	0	4	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	
DES	0	8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	1	0	5	0	0	33	1	69	
DGEO	1	5	0	0	0	0	3	8	0	0	1	1	0	0	0	1	4	0	0	2	0	0	0	0	0	3	0	29	
DH	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	6	0	0	1	0	0	0	0	0	32	0	42		
DL	0	1	1	0	0	0	0	36	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	42		
DTL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	7	30	17	55	
Total	1	17	11	0	1	0	4	51	44	0	1	1	0	0	0	1	11	0	1	32	2	0	7	0	12	187	18	401	

Além disso, o conjunto de disciplinas disponíveis para atribuição em salas de aula foi dividido por departamentos (D), ou seja, a i -ésima disciplina é oferecida pelo DAT, para $1 \leq i \leq 73$, pelo DCC, para $74 \leq i \leq 99$, pelo DPCE, para $100 \leq i \leq 131$, pelo DCJUR, para $132 \leq i \leq 165$, pelo DES, para $166 \leq i \leq 234$, pelo DGEO, para $235 \leq i \leq 263$, pelo DH, para $264 \leq i \leq 305$, pelo DL, para $306 \leq i \leq 347$ e pelo DTL, para $348 \leq i \leq 401$.

Por outro lado, os departamentos concentram a maior parte de suas disciplinas em seus turnos departamentais, e a atribuição de disciplinas fora deles não excede 20% de suas ofertas totais. Esta restrição é incorporada para todos os departamentos, onde os limites superiores são estabelecidos com base no pior cenário, associado à atribuição de, no máximo, 20% das disciplinas de 4 créditos acadêmicos. Seguindo esta premissa, as seguintes restrições foram incorporadas ao modelo:

$$\sum_{i=1}^{73} x_{ijk1m} + \sum_{i=1}^{73} x_{ijk2m} \leq 56, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.25)$$

$$\sum_{i=74}^{99} x_{ijk2m} + \sum_{i=74}^{99} x_{ijk3m} \leq 20, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.26)$$

$$\sum_{i=100}^{131} x_{ijk1m} + \sum_{i=100}^{131} x_{ijk2m} \leq 24, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.27)$$

$$\sum_{i=132}^{165} x_{ijk2m} + \sum_{i=132}^{165} x_{ijk3m} \leq 24, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.28)$$

$$\sum_{i=166}^{234} x_{ijk2m} \leq 52, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.29)$$

$$\sum_{i=235}^{263} x_{ijk2m} + x_{ijk3m} \leq 20, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.30)$$

$$\sum_{i=264}^{305} x_{ijk1m} + x_{ijk2m} \leq 32, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.31)$$

$$\sum_{i=306}^{347} x_{ijk2m} + x_{ijk3m} \leq 32, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4, \quad (5.32)$$

$$\sum_{i=348}^{402} x_{ijk1m} + \sum_{i=348}^{402} x_{ijk2m} \leq 44, \quad j = 1, \dots, 31, \quad k = 1, \dots, 5, \quad m = 1, \dots, 4. \quad (5.33)$$

5.5 Simulações computacionais

5.5.1 Simulação computacional para a gestão operacional do RU-IM

Para realizar simulação com o modelo de alocação de salas de aula, faremos uso do Python, através do Software VsCode, utilizando a biblioteca *CPlex* [29], para auxiliar no processo de modelagem.

A simulação computacional realizada com base nos itens servidos no restaurante universitário do Instituto Multidisciplinar da UFRRJ mostra um cenário do fluxo e demanda das refeições servidas aos estudantes ao longo de uma semana.

Para os resultados das simulações utilizando o modelo da dieta, os dados referentes à quantidade de porções de cada item foram obtidos das tabelas 5.1 e 5.2. No processo, as frutas (maçã, banana, laranja, pera, tangerina) e os pães (pães, bolos, biscoitos) foram tratados como variáveis inteiras. Em contrapartida, as porções dos demais itens fornecidos foram consideradas

como variáveis fracionárias.

Apesar das três refeições diárias oferecidas - café da manhã, almoço e jantar - a simulação focalizou-se no café e no almoço, dado que o cardápio do jantar replica o do almoço.

Ao analisar a tabela 5.4, percebemos que a solução obtida pelo modelo da dieta atende a todas as restrições estabelecidas, incluindo os limites mínimos e máximos de porções para cada alimento. Além disso, garante que a ingestão mínima de proteínas seja de 30 gramas, enquanto a ingestão máxima de carboidratos não ultrapasse 80 gramas. O custo total mínimo para essa refeição é estimado em **28,95 reais**.

Essa composição alimentar resulta em:

- **Café da manhã:** 1 porção de leite, 2 porções de pão, 1 porção de café, 1 porção de achocolatado, 5 porções de bolo de milho, 5 porções de biscoitos, 1 porção de maçã, 1 porção de banana, 1 porção de laranja, 1 porção de tangerina e 1 porção de pera.
- **Almoço/janta:** 3 porções de frango assado, 2 porções de frango grelhado, 2 porções de carne moída, 2 porções de almôndegas, 2 porções de hambúrguer vegetariano, 2 porções de lentilha, 7 porções de arroz, 4 porções de feijão, 7 porções de macarrão, 7 porções de farofa, 1 porção de alface, 1 porção de rúcula, 1 porção de beterraba, 2 porções de grão-de-bico, 1 porção de maçã, 1 porção de banana, 1 porção de laranja e 1 porção de cada um dos refrescos.

5.5.2 Simulação computacional para a gestão de salas de aula do Instituto Multidisciplinar

Para realizar simulação com o modelo de alocação de salas de aula, faremos uso do Python, através do Software VsCode, utilizando a biblioteca *Gurobi* [30], para auxiliar no processo de modelagem.

Com base nos dados coletados no campus da UFRRJ de Nova Iguaçu, incluindo a quantidade e capacidade das salas, número de disciplinas, créditos atribuídos a cada disciplina e os dias da semana, a simulação resultou na geração de uma tabela em formato *.csv* cujo link de acesso está abaixo com o arquivo em PDF de cada departamento. Essa tabela detalha as disciplinas, juntamente com seus créditos, e suas alocações específicas em salas, turnos, horários e dias da semana correspondentes.

Com o intuito de proporcionar uma compreensão mais clara dos resultados, é importante destacar que em Python a contagem dos índices se inicia em 0, ao contrário de 1. Assim, disciplinas, salas, dias, turnos e horários, todos os itens, de uma maneira geral, são contados a partir de 0. O link a seguir apresenta planilhas com as disciplinas alocadas nas salas, horários e turnos em cada departamento do Instituto Multidisciplinar da UFRRJ, como proposto pelo modelo apresentado.

[Resultado Alocação](#)

Alimento	Porções
Leite	1 porção
Pão	2 porções
Cafe	1 porção
achocolatado	1 porção
Bolo de milho	5 porções
Biscoitos	5 porções
maçã	1 porção
Banana	1 porção
Laranja	1 porção
Tangerina	1 porção
Pera	1 porção
Queijos	1 porção
Presunto	1 porção
Geléia	1 porção
Ovo	1 porção
Margarina	1 porção
Manteiga	1 porção
Frango Assado	3 porções
Frango grelhado	2 porções
Carne moída	2 porções
Almondegas	2 porções
HambVeg	2 porções
Lentilha	2 porções
Arroz	7 porções
feijão	4 porções
Macarrão	7 porções
farofa	7 porções
Alface	1 porção
Rúcula	1 porção
Beterraba	1 porção
GrãoDeBico	2 porções
maçãsob	1 porção
Bananasob	1 porção
Laranjasob	1 porção
Guaraná	1 porção
Uva	1 porção
Groselha	1 porção

Tabela 5.4 – Porções de cada alimento

Conclusões e trabalhos futuros

Portanto, como proposto inicialmente neste trabalho, foram realizadas simulações computacionais utilizando dois modelos de programação linear (inteira e mista) em um cenário real. Os resultados foram satisfatórios: no modelo de alocação de salas de aula, foi possível maximizar o número de disciplinas alocadas em suas respectivas salas, além de atender aos alunos matriculados. No modelo da dieta, aplicado ao restaurante universitário do Instituto Multidisciplinar da UFRRJ, alcançamos a minimização de custos sem prejudicar a nutrição dos alunos, viabilizando seu uso em todos os turnos.

É fundamental ressaltar a relevância deste estudo não apenas no âmbito da matemática computacional, mas também na educação. A modelagem matemática desempenha um papel crucial no ensino de disciplinas de nível superior, como geometria analítica, álgebra linear e cálculo. Ela proporciona uma abordagem dinâmica que torna conceitos abstratos mais acessíveis e relevantes para os alunos. A inclusão da modelagem matemática nas salas de aula pode evidenciar como conteúdos de nível médio/superior seriam melhor compreendidos quando relacionados a aplicações reais, aumentando o engajamento dos alunos e preparando-os para enfrentar desafios complexos em diversas áreas do conhecimento.

Paralelamente, está em andamento o desenvolvimento de um modelo multiobjetivo que combina os dois modelos abordados neste estudo. Esse modelo visa aprimorar a gestão em instituições educacionais, como escolas e universidades. Uma vez implementado e validado no Instituto Multidisciplinar da UFRRJ, poderá ser facilmente adaptado a outras instituições de ensino, simplificando a organização das demandas e processos educacionais.

Referências bibliográficas

1

2 M.C. Goldberg and H.P.L. Luna. *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, 2 edition, 2005.

3 G. B. Dantzing. Linear programming. *Operations Research Nutrition Bulletin*, 50(1):42–47, 2002.

4 L. A. Wolsey. *Integer Programming*. Wiley interscience series in discrete mathematics and optimization. John Wiley & Sons, Ltd, 1 edition, 1998.

5 M. Arenales, R. Morabito, V. Armentano, and H. Yanasse. *Pesquisa Operacional: Para cursos de engenharia*. Elsevier, 2015.

6 A. M. Ferreira. Uma abordagem da modelagem matemática para o problema da dieta em restaurantes universitários. Trabalho de conclusão de curso, Instituto Multidisciplinar, Universidade Federal Rural do Rio de Janeiro, Nova Iguaçu, 2021.

7 J.N. Rangel. Uma abordagem da modelagem matemática para o problema de alocação de salas em cursos universitários. Trabalho de conclusão de curso, Instituto Multidisciplinar, Universidade Federal Rural do Rio de Janeiro, Nova Iguaçu, 2023.

8 J. L. Buttriss, A. Briend, N. Darmon, E. L. Ferguson, M. Maillot, and A. Lluch. Diet modelling: how it can inform the development of dietary recommendations and public health policy. *Nutrition Bulletin*, 39:115–125, 2014.

9 C. V. Dooren. A review of the use of linear programming to optimize diets, nutritiously, economically and environmentally. *Frontiers in Nutritions*, 5(48):1–15, 2018.

10 M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. Wiley & Sons, 4 edition, 2010.

11 D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming*. International Series in Operations Research & Management Science. Springer, 4 edition, 2016.

12 R.J. Vanderbei. *Linear Programming: Foundations and Extensions*. International Series in Operations Research & Management Science. Springer, 4 edition, 2013.

- 13 N. A. Baki, N.S.M.N. Mangsor, and M.K.A. Razak. Application of linear programming in students diet problem. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1.5):87–90, 2019.
- 14 A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13:87–127, 1999.
- 15 A. Elloumi, H. Kamoun, B. Jarboui, and A. Dammak. The classroom assignment problem: Complexity, size reduction and heuristics. *Applied Soft Computing*, 14:677–686, 2014.
- 16 G. B. Bucco, C. J. Bornia-Poulsen, and D. L. Bandeira. Desenvolvimento de um modelo de programação linear para o problema da construção de grades horárias em universidades. *Gestão & Produção*, 24(1):40–49, 2017.
- 17 S. Sarin, Y. Wang, and A. Varadarajan. A university-timetabling problem and its solution using benders’ partitioning - a case study. *Journal of Scheduling*, 13:131–141, 2010.
- 18 R. Alvarez-Valdes, E. Crespo, and J. Tamarit. Design and implementation of a course scheduling system using tabu search. *European Journal of Operational Research*, 137:512–523, 2002.
- 19 H. Rudová, T. Müller, and K. Murray. Complex university course timetabling. *Journal of Scheduling*, 14:187–207, 2011.
- 20 E. Burke, J. Marecek, A. Parkes, and H. Rudová. Decomposition, reformulation, and diving in university course timetabling. *Computers & Operations Research*, 37:582–597, 2009.
- 21 S. Daskalaki, T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135, 2004.
- 22 C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Towards improving the utilization of university teaching space. *Journal of the Operational Research Society*, 60(1):130–143, 2009.
- 23 N. Maculan Filho and M.H.C. Fampa. *Otimização linear*. Ed. UnB, 1 edition, 2006.
- 24 A.O. Barboza, F.N. Junior, S.L.V. Bortolotti, and R.A. de Souza. Programação linear inteira mista e algoritmo genético aplicados ao problema de transferência e estocagem de produtos em uma indústria petrolífera. *Revista Eletrônica Sistemas Gestão*, 2015.
- 25 Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research*. McGraw-Hill, New York, NY, 8th edition, 2001.
- 26 Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1998.
- 27 Laurence A Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1998.
- 28 Departamento de Atenção Básica Ministério da Saúde, Secretaria de Atenção à Saúde. *Guia alimentar para a população Brasileira*. Ministério da Saúde, 2 edition, 2014.
- 29 IBM ILOG CPLEX Optimization Studio Documentation. Acesso em: [15/12/2023].
- 30 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.